A systematic overview of data federation systems

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

[a] *KRDB Research Centre, Faculty of Computer Science, Free University of Bozen-Bolzano, Italy*
E-mail: <name>.<surname>@unibz.it
[b] *Department of Information Science and Media Studies, University of Bergen, Norway*
E-mail: guohui.xiao@uib.no
[c] *Department of Informatics, University of Oslo, Norway*
E-mail: guohuix@ifi.uio.no
[d] *Ontopic S.r.l, Italy*
E-mail: <name>.<surname>@ontopic.ai
[e] *Department of Computing Science, Umeå University, Sweden*
E-mail: diego.calvanese@umu.se

Editor: Aidan Hogan, Universidad de Chile, Chile
Solicited reviews: Andriy Nikolov, AstraZeneca, England; Aidan Hogan, Universidad de Chile, Chile; Two Anonymous Reviewers

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 techniques; *(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, deployment, commercial support, open source, and release. Using this framework, we thoroughly studied 51 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

*Corresponding author. E-mail: guohui.xiao@uib.no.

databases to unstructured flat files. According to the prediction by Reinsel et al. [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 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 an interface usually consists of 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. Compared to centralized solutions, though, accessing multiple data sources on the fly renders query answering more challenging [18–20] and requires sophisticated optimization strategies to be devised. 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 [21, 22] and Teiid [23] are two systems supporting respectively SPARQL query answering over multiple SPARQL endpoints (*i.e.*, standardized HTTP services [24] that can process SPARQL queries) and SQL query answering over multiple heterogeneous data sources, like relational databases, structured files and web services. 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* [25–27], *data lakes* [28–31], and *polystores* [32–36], 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) [37] 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 51 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, selected

by following a rigorous and well-founded methodology, comprise 33 industrial systems under active development and with public official documentation, and 18 academic systems. This 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 techniques*.
- 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 five sub-dimensions of *main development language*, *deployment*, *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 51 considered data federation systems, we collect information along 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 [20, 38–43] that reviews the approaches and systems for federated query answering under multiple perspectives. For example, Oguz et al. [20] evaluate seven SPARQL federation query engines by focusing on their query evaluation techniques, while Azevedo et al. [42] study the modern data federation systems (including BigDAWG [33], CloudMdsQL [35], Myria [34], and Apache Drill [44]) 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 51 systems and 4 main evaluation dimensions overall divided into 16 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

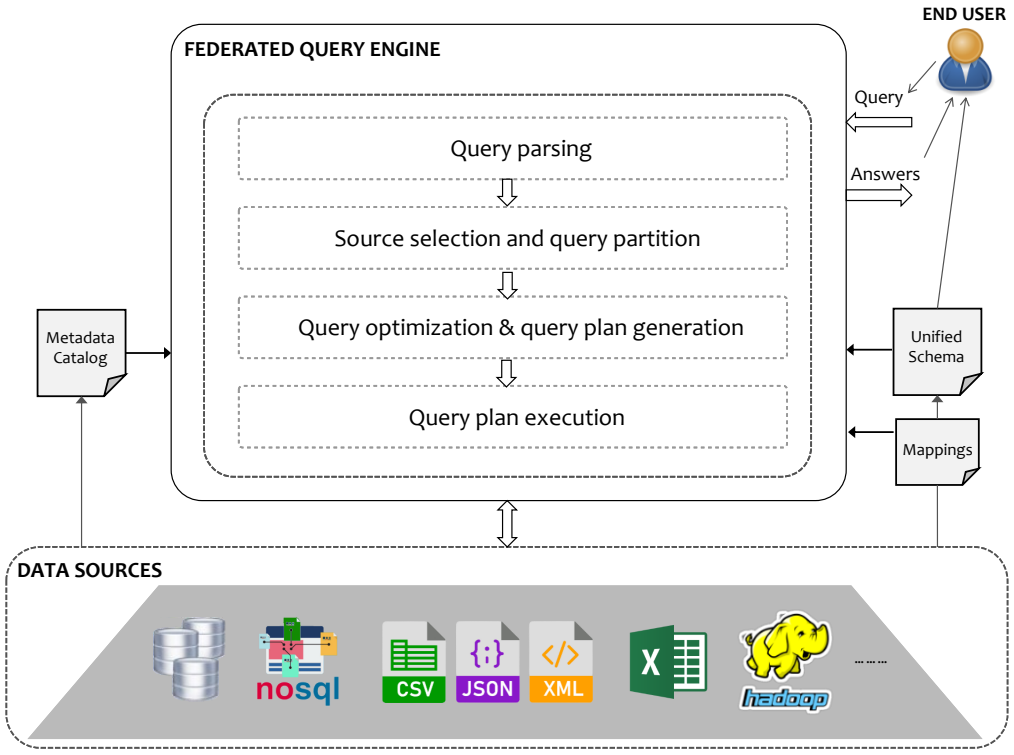

Fig. 1. Typical architecture of a federated query engine (inspired by Oguz et al. [20]).

analyzes the capabilities of these systems according to the proposed framework. Section 7 discusses related work. Section 8 concludes by discussing open problems and challenges as well as giving directions for further work. Appendices A and B respectively provide further details on the specific sources supported by the systems and on our methodology. A Web version of the tables in this paper, including possible corrections and integrations, is available on GitHub.[1]

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.

2.1. Federated query answering

The core task of data federation is federated query answering [20, 38–41]. 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 centralized 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 provide a unified schema of the data sources participating in the federation, map the data in the sources to the unified schema, and provide statistical information about the data sources as well as the information of how these data sources can be accessed. For

[1] https://github.com/ontop/ontop-examples/tree/master/swj-2022-federation-survey

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 warehouses, 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 participating in the federation into the elements conforming to 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. *Query parsing*. This step deals with the syntactic issues of Q, *i.e.*, checking whether the input queries are syntactically correct w.r.t. the adopted query language(s) 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.
2. *Source selection and query partition*. This step selects suitable data sources for each algebraic component of Q, and partitions Q into smaller sub-queries $q_1, \ldots, q_m$ (*i.e.*, query chunks) accordingly, based on the mappings from the data sources to the unified schema $\mathcal{V}$. Approaches for source selection can be index-based, such as the "triple pattern-wise source selection" for SPARQL queries [45, 46], and a way for query partitioning is to try to "push down" the evaluation of the operators to the data sources, rather than perform such evaluation at the level of the federation engine.
3. *Query optimization & query plan generation*. This step computes an execution plan of the partitioned sub-queries $q_1, \ldots, q_m$, 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 may be 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. *Query plan execution*. This step, finally, evaluates the decomposed sub-queries $q_1, \ldots, q_m$ over the corresponding data sources via the mappings and the metadata catalog, and generates the answers of the original query Q. Note that, if the query language that the data source supports 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 clarify the inner workings 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 per 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 NoSQL database 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*, and* :Inspector*, as well as the properties* :hasCode*,* :hasInspector*,* :hasName*, and* :hasSalary*. The set* $\mathcal{M}$ *contains mappings from the data in* S_1 *to the terminology* :Product*,* :hasCode*, and* :hasInspector *of* $\mathcal{V}$*, as well as the mappings from the data in* S_2 *to the terminology* :Inspector*,* :hasName*, and* :hasSalary*.*

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 partitions Q into sub-queries according to some query partition strategy. In our example, by exploiting the mapping set* $\mathcal{M}$*, the federation engine selects source* S_1 *for triple patterns* t_1*,* t_2*, and* t_3*, and selects source* S_2 *for triple patterns* t_4 *and* t_5*. Then, by adopting* exclusive groups*, i.e., a push down strategy for query partition and optimization [21, 22], the engine computes a partition* $\mathcal{Q} = \{q_1, q_2\}$ *of Q, by grouping together the triple patterns corresponding to the same source, so that joins among them are pushed down to the source and a minimal number of federated joins are evaluated.*

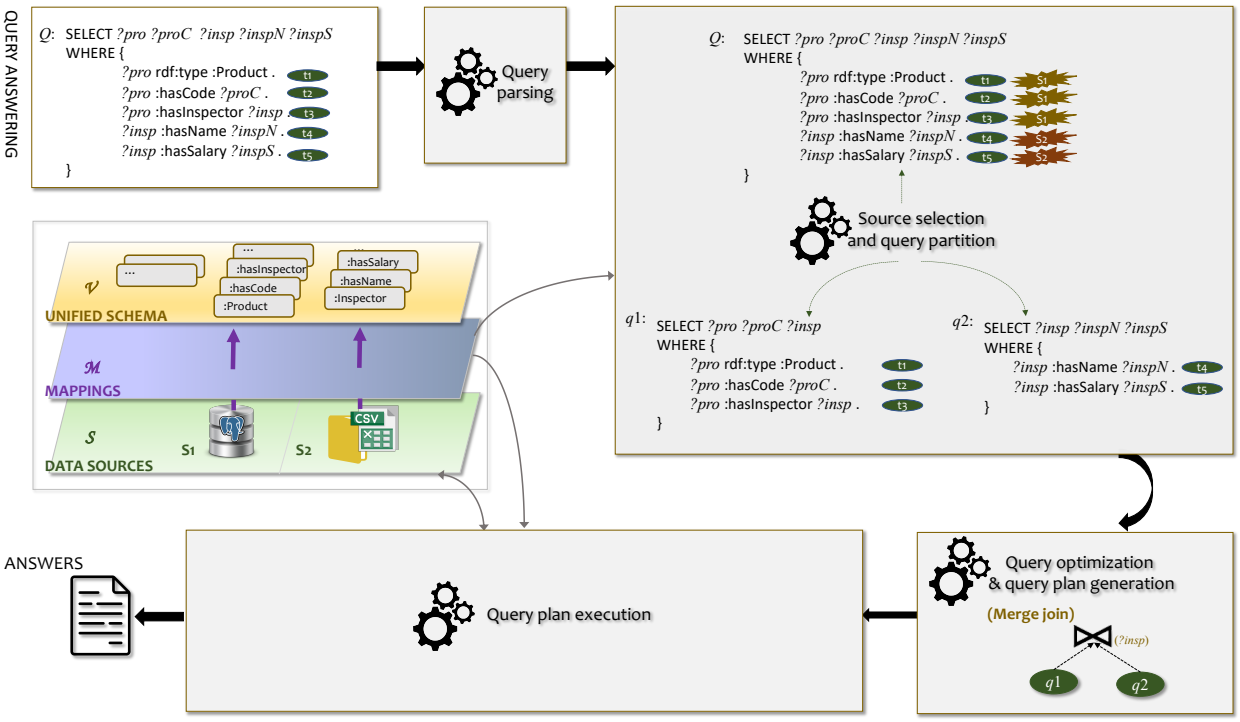

Fig. 2. An example of federated query answering.

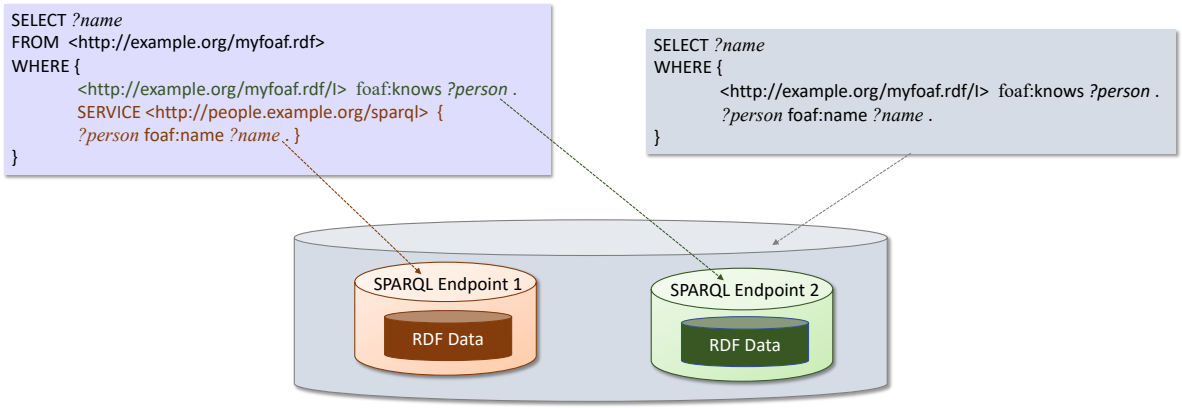

Fig. 3. Example of SPARQL query under the explicit federation setting (left-hand side), and its counterpart under the transparent setting (right–hand side).

After that, the engine computes a plan for evaluating Q. A possible plan is the following: reformulate query q_1 into a SQL query q_1' and query q_2 into a NoSQL query q_2', according to the mappings definitions in $\mathcal{M}$; dispatch q_1' to S_1 and q_2' to S_2, 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.

2.2. Transparent vs explicit federated query answering

From the perspective of whether the data source information is transparent for end users, federated query answering can be classified into *transparent federation* (the one we have discussed so far) and *explicit federation* [9, 45].

Transparent federation gives users the impression to query one single data source despite data being distributed and possibly coming from heterogeneous sources [45]. Hence, it is recognized as a general and ideal[2] solution.

A simplified setting is one where the unified schema is simply a merge of the source schemas, and the user explicitly states in the query the sources against which it should be evaluated. In such a scenario, we talk about explicit data federation. This approach is built-in into SPARQL 1.1 through its dedicated `SERVICE` keyword, and therefore is supported by any SPARQL-based system fully compliant with SPARQL 1.1, including systems not primarily focusing on data federation. Figure 3, left-hand side, shows an example of query formulated under the explicit federation setting, asking for the data from a local RDF store and an explicitly specified remote RDF store. The right-hand side of the same figure shows the same query formulated under the transparent federation setting, assuming that foaf:knows and foaf:name are properties belonging to the unified schema.

Compared with transparent federated query answering, the explicit scenario does not require a procedure of source selection for delivering its task of accessing and joining multiple data sources. However, the burden is placed on end users, and this might constitute a major hindrance in case they are not familiar with the data sources participating in the federation and the data therein contained.

However, the transparent setting is not devoid of drawbacks. For instance, users lose the ability of communicating with specific data sources directly. Moreover, the transparent situation needs to maintain a unified schema mapped to multiple data sources, which means that it is more sensitive to schema updates: when the schema of a source is updated, the unified schema and the mappings may also need to be updated.

2.3. Other capabilities

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 noteworthy capabilities supported by federation systems of this survey.

- *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[3] 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[4] supports SPARQL updates[5] so as to make users able to modify the data of the SPARQL endpoints, and the SQL federation engine Denodo supports SQL data manipulation 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[6] 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 stems from our needs for 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

[2] https://www.w3.org/2009/sparql/wiki/Feature:BasicFederatedQuery#Feature_description
[3] https://community.denodo.com/kb/en/view/document/Denodo%20Security%20Overview
[4] https://rdf4j.org/documentation/programming/federation/
[5] https://www.w3.org/TR/sparql11-update/
[6] https://documentation.sas.com/api/docsets/fedsrvag/4.2/content/fedsrvag.pdf

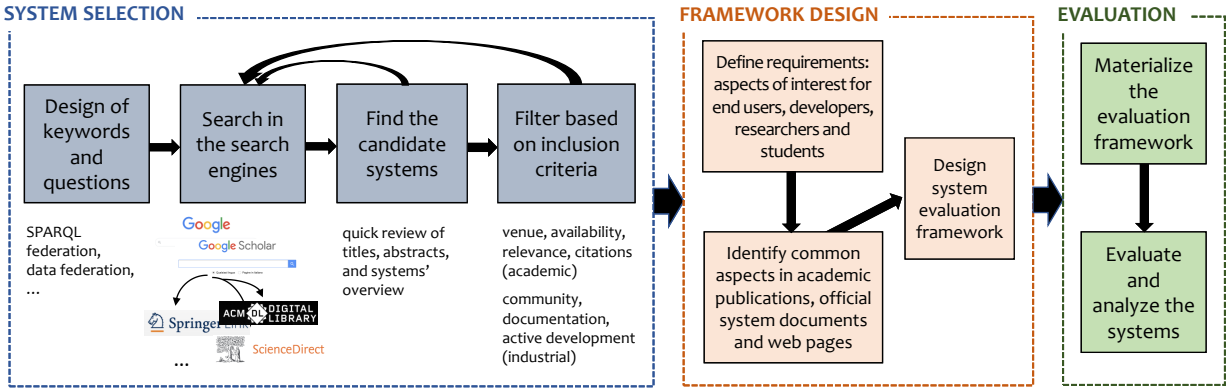

Fig. 4. The overall methodology of the survey work.

(consumers), developers, researchers and students. In this section, we present the overall methodology used for our study. Figure 4 provides a snapshot of our methodology, which consists of the *identification of the considered systems*, the *design of the system evaluation framework*, and the *evaluation of the systems through the framework*.

3.1. The methodology for system selection

As shown in Fig. 4, 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 inclusion criteria*. The bulk of candidate systems collection and filtering required three months, between the end of 2020 and the beginning of 2021. 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 from libraries such as *SpringerLink*, *IEEE Xplore*, *ACM Digital Library*, and so on. By evaluating these titles and abstracts, we selected and downloaded 295 academic publications for further in-depth reading, consisting of papers, technical reports, PhD and master theses whose primary focus is on data federation. They include a majority of *system-specific publications* and 17 *system comparison publications*, which range from data federation surveys to benchmarks, system evaluations and PhD theses reviewing the topic.

After reviewing these publications, we identified a total of 56 academic data federation systems that we narrowed down to a final selection of 18 representative systems based on the following *inclusion criteria*:

- *Scope*. The system must focus on the problem of query federation, or introduce a data federation system.
- *Venue*. The system must be described in *formal publications* such as papers in journals or conference proceedings, and not only in preprints or technical reports.
- *Availability*. The system source code and official website must be available, either linked from the system publications or findable from the authors' GitHub profiles (*e.g.*, the *SPLENDID* system).
- *Relevance*. The system must satisfy at least one of the following criteria: it should be mentioned in system comparison publications; it should provide federation of heterogeneous data sources (*e.g.*, RDBs and CSV files); it should ensure data security.

– *Citations*. For period ⩾2020, we do not consider citations. For period 2015–2019, systems should have at least 10 citations. For each prior period ⩽2008, 2009–2011 and 2012–2014, we only consider the system having the largest number of citations among the ones matching the previous criteria.

The citations criterion aims at keeping the scope of this survey manageable and focused on newer systems, also considering that most systems earlier than 2015 are covered by other system comparison publications and their source code is more likely to be unavailable, making them less interesting to our intended audience. Note that to apply this criterion, a system is classified into the year of its most recent conference or journal publication and a system number of citations is obtained by summing the Google Scholar citations of all its collected publications as of 2022/06/07.

Fine details of the selection process are provided in Appendix B, which reports on: (i) the collected 295 academic publications in terms of aggregated statistics (Section B.1) and full bibliography (Section B.4); (ii) the collected 17 system comparison publications, in terms of metadata, compared systems and considered aspects (Section B.2); and (iii) the selection of 18 systems out of the 56 identified ones, based on attained inclusion criteria (Section B.3).

The selection of industrial systems. To find candidate industrial systems we adopted the Google search engine. We employed the following generic keywords/questions, aiming at including as many systems as possible:

> “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 a data federation system already known by us, like Teiid and Denodo. We collected, deduplicated and reviewed the search results, looking for the websites of industrial data federation systems. Some search results already corresponded to a system website. Others were instead discussing more broadly about data federation/virtualization or recommending/listing/comparing systems referring to data federation, virtualization or integration, in which case we browsed page links to identify any referenced system website. As a result, we collected the official websites of 72 candidate systems that *may provide* (due to noise in search results following the use of generic keywords) the capability of data federation. We then consulted these websites, read the systems descriptions and documentation carefully, and eventually selected 33 industrial systems for our survey work that strictly meet all the following inclusion criteria:

– *Scope*. The system must actually provide the capability of data/query federation.
– *Community*. There should be evidence for a user community around the system, *e.g.*, via usage statistics and user messages in fora, mailing lists, issue trackers and the like.
– *Documentation*. Official system documentation must be publicly available, to support both (perspective) users and ourselves in conducting the analyses reported in this survey.
– *Active development*. There must have been at least a system release since 2015/10, *i.e.*, in the last five year since the time we started this survey (2020/10).

The concrete information of the systems found, as their names, owners, and descriptions, can be found in Section 5.

3.2. The methodology for designing the evaluation framework

To design a framework for evaluating data federation systems in a uniform and qualitative way, also considering the intended audience of this survey, we focus on answering the following question (see *framework design* in Fig. 4):

> *What aspects of data federation systems are relevant for end users, developers and scholars?*

While in principle answers can be obtained by interviewing these three groups (*e.g.*, via questionnaires), this approach presents two main difficulties: (i) it is hard to identify a representative sample to interview; and (ii) it is hard for interviewees to answer the question in a general and comprehensive way. Instead, we rely on the fact that data

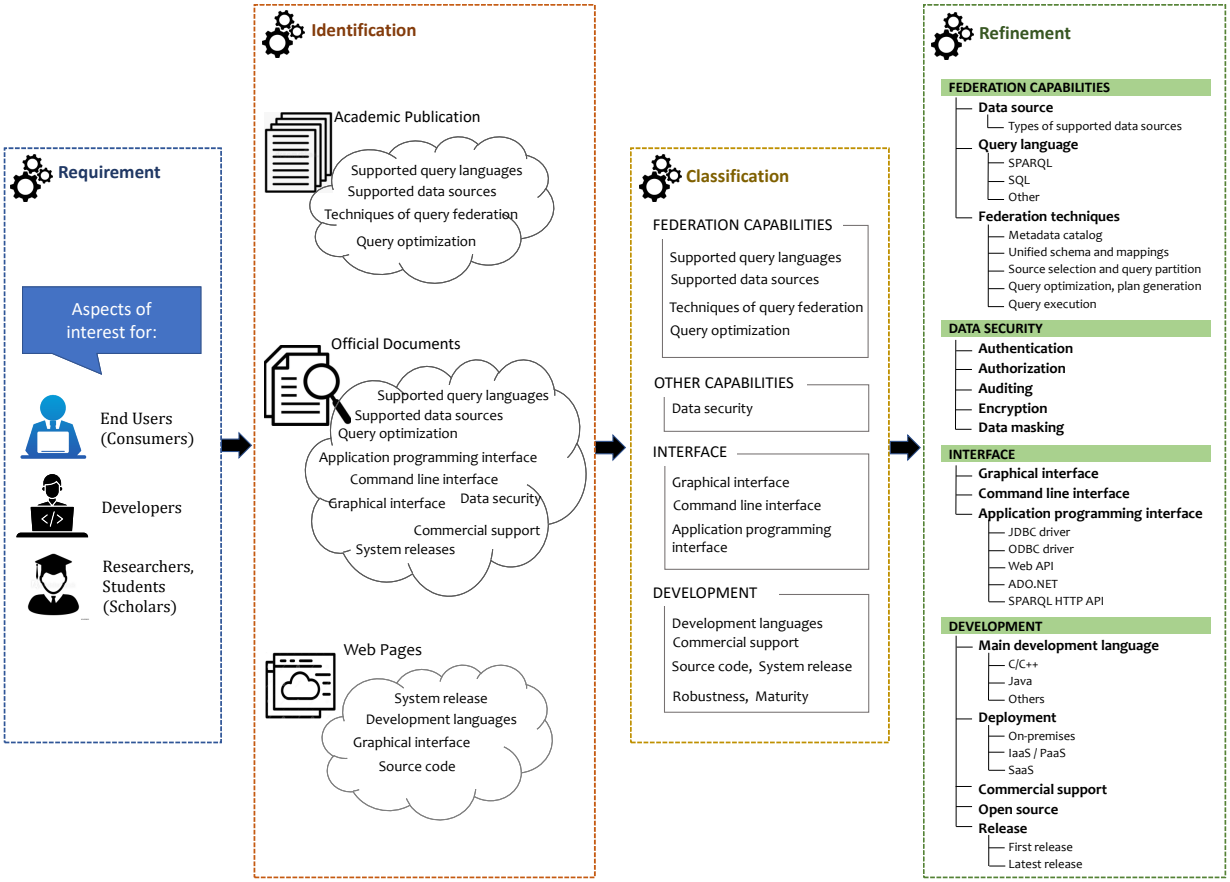

Fig. 5. The generation of the system evaluation framework.

federation is an established domain that has been studied for decades in both the Database and the Semantic Web communities, leading to a large body of information from which to extract the aspects of interest that answer our question. Concretely, we consider three information sources:

- *Academic publications*. We look for aspects deemed important by other surveys on data federation, or that are frequent in academic publications referring to data federation systems.
- *Official documents*. We look for aspects commonly present in official documents of data federation systems, such as user and developer guides.
- *Web pages*. We look for aspects that are often considered when comparing data federation systems.[7]

The system evaluation framework, consisting of four dimensions with sub-dimensions, is generated by combining, classifying, and further refining the identified aspects. The full process is depicted in Fig. 5. Starting from our original question (*Requirement* box), we report the "raw" aspects identified in academic publications, official documents and web pages (*Identification* box). They were classified into four categories (*Classification* box), which then underwent a series of refining steps (*Refinement* box), guided both by the information sources we reviewed and by our own expertise as researchers and developers, as well as our own experience on the data federation task and systems. This refinement results in a final evaluation framework that addressess the information needs of the different audience groups targeted by this survey:

[7] *E.g.,* https://en.wikipedia.org/wiki/Comparison_of_relational_database_management_systems

- *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 a graphical interface to help them to set up a data federation instance easily, whether it provides the services for solving the problems they may encounter, whether it provides the techniques for protecting their data from leakage, and whether it is robust enough so as to withstand the technical difficulties that may be encountered in production (*e.g.*, load spikes, temporary source unavailability, etc).
- *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, *e.g.*, to support more complex data consumption scenarios.
- *Researchers and students*. They conduct research or studies on data/query federation. Thus, the aspects of interest for them relate to the knowledge of the capabilities of the systems, or of the strategies they adopt.

All of the aforementioned aspects of interest are captured by the dimensions and sub-dimensions of our evaluation framework, as it will be detailed in Section 4.

3.3. The methodology of system evaluation

After identifying the considered systems and the evaluation framework, we use such 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. 5, consists of four dimensions: *federation capabilities*, *data security*, *interface*, and *development*. Each dimension is further characterized by sub-dimensions (16 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 techniques.

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 it will preferably take into consideration systems supporting CSV files and MySQL at the same time. This sub-dimension also permits users to distinguish whether a system focuses on heterogeneous or homogeneous data sources. Roughly speaking, the more different types of data sources a system supports, the more capable that system is in accessing heterogeneous data. 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 federation system should preferably adopt a standard query language that is familiar to most people, like SPARQL or SQL. 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. We considered the systems developed within the Semantic Web and Database communities, but not limited to these two kinds. Hence, we characterize this sub-dimension into *SPARQL*, *SQL*, and *Other*.

Federation technique sub-dimension. We refer to the typical architecture for federated query answering described in Section 2, and assess the main techniques adopted by a system. We focus on the techniques for *metadata catalog*, *unified schema and mappings*, *source selection and query partition*, *query optimization and plan generation*, and *query execution*. The motivation is to help readers in forming a general idea about the techniques employed by each system.

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, or 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[8] is the process of masking (*i.e.*, obscuring, 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). Data masking may use lossless techniques such as encryption or tokenization[9] that allow retrieving the original unmasked value if the required information is available (*e.g.*, the decryption key), but this feature is not a requirement and is not provided by many masking approaches that just aim at hiding sensitive data (*e.g.*, the simple replacement of data with random values, or with '$*$' characters). Also, differently from encryption that may operate on the whole communication channel between the system and the user, data masking typically operate on the individual pieces of sensitive data (*e.g.*, a table column or row field).

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.

[8] https://en.wikipedia.org/wiki/Data_masking
[9] https://en.wikipedia.org/wiki/Tokenization_(data_security)

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[10] 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.

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 influence system requirements (*e.g.*, a Java Runtime Environment is required for the Java language), performance, customization, and integration options (*e.g.*, embedding the system as a library), and consequently affect the system's fitness for use in an intended user application.

Deployment sub-dimension. The hardware/software infrastructure required to run a system, as well as its economic viability, are influenced by the deployment options offered for the system. At one end of the spectrum, we have *on-premises* deployment where the user obtains the software, possibly for a one-time license fee, and is in charge of its deployment, maintenance (*e.g.*, updates) and configuration, which may occur on any machine(s) under the user control (*i.e.*, "on the premises" of the user). The other end of the spectrum is represented by *Software as a Service* (SaaS),[11] whereby the system is offered as a pre-deployed service maintained by the provider, and the user only cares about configuring and using the service on a subscription basis, where costs may depend on "how long" (*e.g.*, hours) and "how much" (*e.g.*, number of queries, transfered data) the service is used. In between, *Infrastructure as a Service* (IaaS) and *Platform as a Service* (PaaS) are intermediate options where system deployment and maintenance are up to the user (as for on-premises deployment), but the system comes bundled with infrastructural resources, such as virtual machines or middleware, of a cloud provided (*e.g.*, Amazon AWS, Microsoft Azure, Google Cloud Platform), these resources managed to different extents by the user (IaaS) or the provider (PaaS). Examples are container platforms like Kubernetes or OpenShift, or cloud marketplaces where virtual machines pre-configured with the system are obtained and subscription fees are divided among system and infrastructure providers.

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 the form of training, timely bug fixes, and installation and customization services, plays a keys role when choosing a system.

[10] https://teiid.github.io/teiid-documents/master/content/reference/r_xml-deployment-mode
[11] https://en.wikipedia.org/wiki/Cloud_computing#Service_models

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 here considered (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 become more familiar 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 (33 in total) was mainly extracted from their official websites, while for the academic systems (18 in total), information was mostly extracted and summarized from their academic publications, although we also considered their online documentation if available.

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 databases, 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 (SaaS).

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 per their latest version (last update on November 20$^{\text{th}}$, 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 of Appendix A.

Table 1

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

System	Provider	Description
AllegroGraph [47]	Franz Inc.	Distributed graph & document DB supporting OWL, SPARQL, SHACL and federation
Amazon Athena [48]	Amazon.com, Inc.	Inter. cloud query service for Amazon S3 data, based on Presto [49]
Amazon Neptune [50]	Amazon.com, Inc.	Fully-managed cloud graph DB (property graph, RDF), part of Amazon AWS
AnzoGraph DB [51]	Cambridge Semantics	Massively-parallel distributed graph DB (property-graph, RDF) for large-scale analytics
Apache Drill [44, 52]	Apache Software Foundation	Distributed schema-free engine for interactive SQL queries on heterogeneous & nested data, inspired by Dremel [53]
Apache Jena [54]	Apache Software Foundation	SPARQL query engine of Jena framework and TDB triple store, supporting federation
Apache Spark [55, 56]	Apache Software Foundation	Multi-lang. (incl. SQL) distributed engine for large-scale data processing & analytics
BigDAWG [33, 57]	Intel Science & Technology Center for Big data	Polystore with heterogeneous storage engines for time series (SciDB), text (Accumulo) and relational data (PostgreSQL)
Blazegraph [58]	Systap	Triple store supporting SPARQL 1.1 federation and powering Wikidata (via a fork)
CloudMdsQL [35, 59]	Inria & LIRMM	Polystore integrating heterogeneous storage engines (incl. RDBMS, NoSQL, HDFS)
Comunica [60]	Univ. Ghent	Modular JS federated query engine for heterogeneous web sources, incl. SPARQL endpoints
CostFed [61]	Univ. Leipzig	Index-assisted, cost-based data federation system for SPARQL endpoints
DARQ [45]	Univ. HU Berlin	Earliest data federation system for SPARQL endpoints, cost-based
Data Virtuality [25]	Data Virtuality GmbH	Heterogeneous data integration solution combining data virtualization and ETL
Denodo [62]	Denodo Technologies Inc.	Data virtualization solution for heterogeneous sources, also available as cloud service
Dremio [63]	Dremio Corporation	Data “lakehouse” (lake + warehouse) solution supporting heterogeneous data sources
FEDRA [64]	Univ. Nantes (LINA lab.)	Data federation system for SPARQL endpoints exploiting data replication
FedX (RDF4J) [21, 22]	fluid Operations AG	On-demand (no statistics, query-time) data federation system for SPARQL endpoints
GraphDB [65]	Ontotext	Triple store featuring OWL reasoning, SPARQL federated queries & RDBMS access
HiBISCuS [66]	Univ. Leipzig	Source selection for SPARQL data federation (DARQ, FedX & SPLENDID extension)
IBM Cloud Pak for Data [67]	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 [68]	IBM	Massively-parallel Hadoop SQL engine for heterogeneous sources (formerly IBM SQL)
IBM InfoSphere Federation Server [69]	IBM	SQL-based data federation system for heterogeneous sources (formerly WebSphere Federation Server)
JBoss Data Virtualization [70]	Red Hat, Inc.	Data federation system based on Teiid and providing read/write access to heterogeneous sources, data security, and multiple user interfaces / APIs
Metaphactory [71, 72]	metaphacts GmbH	KG platform on top of SPARQL endpoints with two federation engines (Ephedra, FedX)
Myria [34]	Univ. Washington	Cloud service for big data management/analytics with parallel & federated query engine
Neo4j (Fabric) [73]	Neo4j, Inc.	Federation solution of Neo4J graph DB (Cypher [74] queries on property graph model)
Obi-Wan [75, 76]	Inria & Polytechnic Institute of Paris	Ontology-Based Data Access (OBDA) [77] system on top of Tatooine [78] mediator for heterogeneous sources
Odyssey [79]	Univ. Aalborg & Univ. Nantes	Statistics & cost-based optimizer for SPARQL data federation (FedX extension)
Ontario [28]	L3S Research Center	Heuristics-based system using RDF Molecule Templates (introduced by its predecessor MULDER [80]) to describe/map source content as star-shaped RDF instance descriptions
Onto-KIT [81]	Univ. Toulouse	Data federation system focusing on Earth Observation data with hypergraph-based data model and query processing techniques
Oracle Big Data SQL [82]	Oracle Corporation	Data federation system for Oracle DB that accesses Hadoop storage & processing
Oracle DB (Spatial & Graph) [83, 84]	Oracle Corporation	Oracle DB component for semantic technologies with data federation capabilities (RDF views) over relational, graph, and RDF (SPARQL) sources
PolyWeb [32, 85]	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 [49, 86]	Presto Foundation	SQL-based distributed query engine suitable to interactive (big) data analytics
Querona Data Virtualization [87]	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
RDFLib [88]	RDFLib team	A pure Python package for working with RDF, supporting SPARQL 1.1 federation
SAFE [89]	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
SAGE [90]	Univ. Nantes	SPARQL engine with “web preemption” (i.e., query suspend/resume) & federation capabilities
SAP HANA [91]	SAP SE	In-memory DB targeting with data federation capabilities, also available as cloud service
SAS Federation Server [92]	SAS Institute	Data federation system featuring data caches, masking, encryption & quality functions
SemaGrow [93]	IIT NCSR ‘Demokritos’	Data federation system for SPARQL endpoints with statistics-based query optimization
SPLENDID [46]	Univ. Koblenz-Landau	Data federation system for SPARQL endpoints that provide VOID [94] data statistics
SQL Server (PolyBase) [95]	Microsoft Corporation	SQL Server component for data federation supporting Hadoop and Azure storage
Squerall [96]	Univ. Bonn	Data federation system for heterogeneous sources built on Spark, Presto, and RML mappings
Starburst [97]	Starburst Data, Inc.	Commercial distribution of Trino, extra security features, available on-premise/on-cloud
Stardog [98]	Stardog Union	KG platform including data federation of heterogeneous sources & query-time inference
Teiid [23]	Red Hat, Inc.	SQL-based engine for data federation of heterogeneous sources
TIBCO Data Virtualization [99]	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 [100]	Trino Software Foundation	SQL-based query distributed engine for interactive big data analytics, forked from Presto
Virtuoso [101–103]	OpenLink Software	Multi-model DB (object-relational, RDF, XML) with data federation facilities

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	✓	–	–	–	✓	–	–	–	–
Apache Spark	–	✓	–	✓	–	–	✓	–	–
BigDAWG	–	–	BigDAWG Query	✓	–	✓	–	–	✓
Blazegraph	✓	–	–	–	✓	–	–	–	–
CloudMdsQL	–	–	CloudMdsQL	✓	✓	✓	–	–	–
Comunica	✓	–	GraphQL	–	✓	–	✓	–	–
CostFed	✓	–	–	–	✓	–	–	–	–
DARQ	✓	–	–	–	✓	–	–	–	–
Data Virtuality	–	✓	–	✓	✓	✓	✓	✓	✓
Denodo	–	✓	GraphQL	✓	–	✓	✓	✓	✓
Dremio	–	✓	–	✓	–	✓	✓	–	–
FEDRA	✓	–	–	–	✓	–	–	–	–
FedX (RDF4J)	✓	–	–	–	✓	–	–	–	–
GraphDB	✓	✓	Cypher	✓	✓	–	–	–	–
HiBISCuS	✓	–	–	–	✓	–	–	–	–
IBM Cloud Pak for Data	–	✓	–	✓	–	✓	✓	✓	✓
IBM Db2 Big SQL	–	✓	–	✓	–	✓	✓	–	✓
IBM InfoSphere Federation Server	–	✓	–	✓	–	–	✓	✓	✓
JBoss Data Virtualization	–	✓	–	✓	–	✓	✓	✓	✓
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	–	✓	–	✓	–	✓	✓	–	✓
RDFLib	✓	–	–	–	✓	–	–	–	–
SAFE	✓	–	–	–	✓	–	–	–	–
SAGE	✓	–	–	–	✓	–	–	–	–
SAP HANA	–	✓	–	✓	–	–	–	–	✓
SAS Federation Server	–	–	FedSQL	✓	–	–	–	–	✓
SemaGrow	✓	–	–	–	✓	–	–	–	–
SPLENDID	✓	–	–	–	✓	–	–	–	–
SQL Server (PolyBase)	–	✓	–	✓	–	✓	✓	–	–
Squerall	✓	–	–	✓	–	✓	✓	–	–
Starburst	–	✓	–	✓	–	✓	✓	–	✓
Stardog	✓	–	–	✓	✓	✓	✓	–	✓
Teiid	–	✓	–	✓	–	✓	✓	✓	✓
TIBCO Data Virtualization	–	✓	–	✓	–	✓	✓	✓	✓
Trino	–	✓	–	✓	–	✓	–	–	✓
Virtuoso	✓	✓	–	✓	✓	–	–	–	–
Number	**27**	**22**	**10**	**32**	**30**	**24**	**24**	**10**	**20**

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

1. With no significant distinction between industrial or academic systems , the standard and popular query languages SQL and SPARQL are adopted by most of these systems to query the data involved in the federation. This choice definitely eases the integration of the system with other possible interacting applications. 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 [74] 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. Some of the academic SPARQL-based systems support only BGP-like queries, such as Obi-Wan [75] and Squerall [96]. Other systems support general SPARQL queries but their publications only discuss federation techniques tailored towards BGPs, such as CostFed [61], HiBISCuS [66], Ontario [28], PolyWeb [32], SAFE [89], SemaGrow [93] and SPLENDID [46]. General SPARQL support may be achieved by relying on a fully-fledged SPARQL engine like RDF4J[12] (formerly Sesame) that supports further operators such as `UNION` and `OPTIONAL`.
4. For systems supporting SPARQL federation, only a few systems, like *Amazon Neptune* and *Apache Jena*, provide the capability of explicit query federation via the `SERVICE` keyword. Among non-SPARQL systems, only *CloudMdsQL* does not support transparent federation.

Data source. Uniformly evaluating and analyzing systems in terms of supported data sources is a challenging task for several 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,[13] while Denodo[14] distinguishes between the classes of *JDBC databases*, *ODBC sources*, and *multidimensional databases*. Instead, Apache Drill[15] and Trino[16] list all the data sources they support without any classification, and IBM Cloud Pak for Data Virtualization[17] solely classifies the supported data sources into IBM data sources, third-party data sources, and files. Secondly, systems may list as supporting 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 simply the only ones supported through the interface, which we thus disregard in our assessment. Finally, sources not supported *directly* by a system, may be supported *indirectly* by combining the system with a suitable third-party adapter component, such as a SQL connector exposing a non-relational data source (e.g., MongoDB) through a standard relational interface (e.g., JDBC), as further discussed in Appendix A. Since such combinations are potentially limitless and the feasibility of each should be assessed (*e.g.*, to verify whether combined components are actually compatible), we here consider only directly supported sources and further discuss the issue in Section 6.5. Overall, all the aforementioned factors make it difficult to assess the supported data source sub-dimension uniformly and precisely.

[12] https://rdf4j.org/

[13] https://teiid.github.io/teiid-documents/master/content/reference/r_data-sources.html

[14] https://community.denodo.com/docs/html/browse/8.0/en/vdp/vql/generating_wrappers_and_data_sources/creating_data_sources/creating_data_sources

[15] https://drill.apache.org/docs/connect-a-data-source-introduction/

[16] https://trino.io/docs/current/connector.html

[17] https://www.ibm.com/docs/en/cloud-paks/cp-data/4.5.x?topic=data-supported-sources

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) relational query engines, and distributed/cloud relational 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" [104], ranging from opaque values to arbitrarily complex nested documents;[18] *(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 Twitter one); and *(vi) Other*. We manually classified each *occurrence* of a specific data source (*e.g.*, MySQL, MongoDB) among the ones supported by a system, under one of the considered 6 data source types (*e.g.*, relational and aggregate-oriented, respectively), depending on how the specific source is accessed by the system and also relying on established system classifications (*e.g.*, DB-Engines [105] and Database of Databases [106] 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 data) and protocols (*e.g.*, IMAP), and various specialized web APIs. We remark that source classification is not *global* across systems but rather *local* to each data federation system supporting that source, so for instance a multi-model database like Virtuoso (when used as a source) may be classified as *relational* if accessed via SQL, or as *graph-based* if access occurs via SPARQL.

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

1. Industrial systems usually support more data sources than academic systems (respectively, 3.2 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. This dominant role, along with the generality and well-understood semantics of the relational model, might also partially explain the proliferation of SQL *connectors/adapters* for non-relational data sources (see discussion in Appendix A). Such proliferation facilitates, for a data federation system supporting the connector/adapter data access interface (*e.g.*, JDBC), extending the support to additional, unanticipated data sources.
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 (24 out of 51, *i.e.*, 47%). 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 51, *i.e.*, 47%) similar to the one for graph-based sources and structured files, and are present both in industrial systems (18 out of 33, *i.e.*, 55%) and, to a lesser degree, in academic systems (6 out of 18, *i.e.*, 33%).
5. Web service paradigms, although important (many sources are available only as web services), are considered less often (10 systems out of 51, *i.e.*, 20%). 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 33, *i.e.*, 55%) more than academic systems (2 out of 18, *i.e.*, 11%).
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 (5 out of 22, *i.e.*, 23%). Conversely, systems supporting SPARQL queries focus on graph-based sources (25 systems out of 27, *i.e.*, 93%) but support relational sources more frequently (10 out of 27, *i.e.*, 37%) than SQL systems do with graph-based sources.

Table 3

Summary of the main techniques used in federated query answering, grouped by affected main component of a typical data federation system. For each technique, we provide references to the literature describing the technique, as well as example systems known to implement the technique

	Techniques
Metadata catalog	**Automatic collection of source metadata** (*e.g.,* data summaries [32, 61])
	Example Systems: AnzoGraph DB, Apache Spark, *CostFed*, Data Virtuality, Denodo, Dremio, FedX(RDF4J), *HiBISCuS*, IBM Cloud Pak for Data, JBoss Data Virtualization, Neo4j(Fabric), *Odyssey*, *Ontario*, Oracle Big Data SQL, *PolyWeb*, Presto, *SAFE*, SAP HANA, SAS Federation Server, *SemaGrow*, *SPLENDID*, Starburst, Stardog, Teiid, TIBCO Data Virtualization, Trino
	Manual provision of source metadata (*e.g.,* VoID [94], Sevod [107], Service descriptions [45])
	Example Systems: Apache Drill, *DARQ*, Denodo, JBoss Data Virtualization, Oracle Big Data SQL, Presto, Querona Data Virtualization, *SemaGrow*, *SPLENDID*, Starburst, Teiid, Trino,
Unified schema and mappings	**Simple merge of schemas**
	Example Systems: Comunica, *CostFed*, *DARQ*, *HiBISCuS*, Neo4j(Fabric), *Odyssey*, *Ontario*, *SAFE*
	Configurable unified schema (*e.g.,* virtual databases [23])
	Example Systems: AnzoGraph DB, Apache Spark, Data Virtuality, Denodo, Dremio, IBM Cloud Pak for Data, JBoss Data Virtualization, *Onto-kit*, Oracle Big Data SQL, *PolyWeb*, Querona Data Virtualization, SAP HANA, SAS Federation Server, *Squerall*, Stardog, Teiid, Trino
Source selection and query partition	**Index-based** [32, 46, 66, 89, 108]
	Example Systems: CostFed, *DARQ*, *FEDRA*, *HiBISCuS*, *Ontario*, *PolyWeb*, *SAFE*, *SPLENDID*
	Query-based [22, 32, 46, 66, 89]
	Example Systems: FEDRA, FedX(RDF4J), *HiBISCuS*, *PolyWeb*, *SAFE*, *SPLENDID*
	Graph-based [61, 66, 81, 89]
	Example Systems: CostFed, *HiBISCuS*, *Onto-kit*, *SAFE*
	Push down [109, p. 326][110–115]
	Example Systems: Data Virtuality, IBM Db2 Big SQL, SQL Server (PolyBase), Starburst, Teiid, Trino
Query optimization and query plan generation	**Cost-based optimization** [116]
	Example Systems: Apache Drill, Apache Spark, *CostFed*, *DARQ*, Data Virtuality, Denodo, Dremio, IBM Cloud Pak for Data, Neo4j(Fabric), *Odyssey*, Presto, SAP HANA, *SemaGrow*, SQL Server (PolyBase), Starburst, TIBCO Data Virtualization, Trino
	Rule-based optimization [21, 28, 32, 117, 118]
	Example Systems: Data Virtuality, FedX(RDF4J), *Ontario*, *PolyWeb*, SAP HANA, Teiid
	Materialization [109, §13.5]
	Example Systems: Denodo, IBM Db2 Big SQL, JBoss Data virtualization, Starburst, Teiid
Query execution	**Bind join** [109, p. 166][20, 45, 93]
	Example Systems: CostFed, *DARQ*, Data Virtuality, *FEDRA*, FedX(RDF4J), *PolyWeb*, *SAFE*, *SemaGrow*, *SPLENDID*, Teiid
	Nested loop join [20, 116]
	Example Systems: Apache Drill, *DARQ*, Data Virtuality, Denodo, *FEDRA*, FedX(RDF4J), *PolyWeb*, *SAFE*, TIBCO Data Virtualization
	Hash join [20, 116]
	Example Systems: Apache Drill, Apache Spark, *CostFed*, Denodo, *SemaGrow*, *SPLENDID*, TIBCO Data Virtualization
	Merge join [116]
	Example Systems: Apache Drill, Apache Spark, Data Virtuality, Denodo, *SemaGrow*, TIBCO Data Virtualization
	Broadcast join [119–121]
	Example Systems: Amazon Athena, Apache Drill, Apache Spark, Starburst, Trino
	Partitioned (shuffle) join [122–124]
	Example Systems: Apache Spark, BigDAWG, Presto, Trino
	Semijion [116]
	Example Systems: TIBCO Data Virtualization
	Parallelization [109, §8.4][20]
	Example Systems: AllegroGraph, Amazon Athena, Apache Drill, Data Virtuality, Denodo, FedX(RDF4J), Squerall, Starburst, TIBCO Data Virtualization
	Data Movement/Ship [125]
	Example Systems: Denodo, TIBCO Data Virtualization
	Caching [20, 22, 122]
	Example Systems: Apache Drill, Apache Spark, *Comunica*, Data Virtuality, Denodo, Dremio, *FEDRA*, FedX(RDF4J), *HiBISCuS*, IBM Cloud Pak for Data, IBM Db2 Big SQL, JBoss Data Virtualization, Presto, *SAFE*, SAP HANA, SAS Federation Server, Teiid, TIBCO Data Virtualization, *DARQ*

Federation techniques. Besides the supported query languages and data sources, we also considered the specific techniques used by each of the selected systems. Table 3 organizes such techniques according to the main components of a typical data federation system as shown in Fig. 1. Note that the categories *Unified schema and mappings* and *Source selection and query partition* are only suitable for transparent federation. For each technique, we provide references to the literature and a list of systems for which the adoption of such technique is stated in official documents or publications. Hence, the lack of the indication of a particular system under a particular technique has to be interpreted as unavailable information, and not as negative information. This holds true especially for closed-source industrial systems, where information about these technical aspects is often covered scarcely or not covered at all in systems' documentation. We next discuss each element of Table 3.

- *Metadata catalog*. A fundamental classification of federation techniques for this component distinguishes between techniques where the metadata catalog is automatically built out of source metadata accessed in a standard way (*e.g.*, via the SQL "Information Schema"), and techniques that allow for manual provision of such metadata by users. These technique families are complementary and a system may adopt one or both of them (*e.g.*, Denodo, see Table 3 for other examples). Manually supplied metadata may be described through self-defined dialects, such as the XML syntax of Teiid and the RDF molecule template of Ontario. Alternatively, some systems adopt standard languages, such as the VoID [94] vocabulary for Linked Data [126] (*e.g.*, Squerall) or the SQL extension for the "Management of External Data", SQL/MED [127] (*e.g.*, Data Virtuality and Teiid, in alternative to its own XML). SQL/MED provides specialized SQL data definition language (SQL DDL) statements, such as `CREATE FOREIGN TABLE`, for defining the objects stored in the federated sources and how to access them. In place of SQL/MED, other systems (*e.g.*, Apache Spark) use regular or customized versions of plain SQL DDL statements, such as `CREATE TABLE` with additional clauses, for the purpose of acquiring catalog metadata and without the intent of actually modifying the source itself.
- *Unified schema and mappings*. We divide the federation techniques for this component into two families: the one where the virtual schema is simply a merge of all the source schemas, and the one where the virtual schema is fully customizable by the user. In Table 3, many examples of the former category are SPARQL-based systems that federate SPARQL endpoints, while most of the examples of the latter category are either systems such as *PolyWeb* that allow the definition of a flexible virtual schema through R2RML/RML mappings, or SQL-based systems that allow the definition of views over the source data, as well as constraints over such views (*e.g.*, primary and foreign keys).
- *Source selection and query partition*. A common approach for the identification of the sources of a query relies on the pre-computation of an index out of the information available in the metadata catalog. Another technique involves the evaluation of probing queries and is exemplified by many SPARQL-based systems in Table 3. One of them is FedX (RDF4J), which issues a probing SPARQL ASK query for each triple pattern in the input query, so as to dynamically identify non-empty sources for that pattern in a more precise, albeit slower, way than using the index. Some systems, like *SPLENDID*, combine these two approaches to gather their respective strengths. Other systems, like *HiBISCuS*, propose a refinement of the query-based strategy where the candidate sources identified by the probing queries are further pruned through an analysis based on the structure of the SPARQL query. For SQL-based systems, source selection is straightforward in the typical scenario where tables of the unified schema are mapped 1:1 to their respective sources, but becomes non-trivial when table data is contributed by multiple sources, as it occurs with data partitioning or replication. Teiid "multisource models",[19] for instance, support horizontal table partitioning across sources (*e.g.*, an employee table partitioned across departments) by defining a source-denoting column (*e.g.*, the department name) in the unified table schema, and exploiting `WHERE` conditions on that column to select a subset of sources to answer the query. For both SPARQL- and SQL-based systems, once sources are identified, query partitioning into sub-queries may involve the push down of query operators to those sources supporting them. An example is the push down of join operators to RDBMS sources [109, p. 326], a technique pioneered in the Garlic system [110].

[18] 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).

[19] https://teiid.github.io/teiid-documents/master/content/reference/r_multisource-models.html

– *Query optimization and query plan generation*. Some systems rely on fully-fledged cost models for generating an optimized query plan, as per the traditional setting of query answering against a single relational datasource. This plan also indicates the evaluation order of sub-queries and the types of joins to be used to combine their results. In other systems, like Ontario, the optimization is purely driven by heuristics and optimization steps are performed according to a pre-defined set of deterministic rules, such as pushing down certain operators (*e.g.*, selection, projection) as much as possible to reduce the size of intermediate results. Cost-based and rule-based optimization may be also combined to attempt generating better query execution plans, as done for instance by Data Virtuality and SAP HANA. Finally, a complementary technique is the creation of materialized views, which can be used in place of re-computing each time the result of expensive distributed operations, in those scenarios where the source data is expected not to change frequently.
– *Query execution*. Apart from standard join techniques such as nested loop or hash join, data federation systems provide techniques for query plan execution that are specifically tailored towards the federated setting. A common trait of these techniques is that they aim at minimizing *data movement* across the different systems participating in the federation. In the *bind join* between two relations, the outer relation is sequentially scanned for join values, which are then used to "bind" the attributes in the inner relation. For each such bind, the matching tuples in the inner relation are transferred to the source of the outer relation and used to construct the result. This approach can be seen as multiple application of the *semijoin* technique, where one side of the join is first filtered with the matching values, and then this "reduced" relation is sent to the other source for performing the actual join. The *broadcast join*, instead, "broadcasts" the matching tuples of the inner relation to all sources in the federation, which is an effective strategy when the outer relation is spread across several sources and the inner relation is much smaller than the outer relation. Splitting relations into smaller chunks lies at the basis of the *partitioned join*, where relations are partitioned according to values of the join keys. This join technique works in combination with *parallelization*, where computation is performed in a distributed way across multiple nodes at the same time. Finally, *caching* of the intermediate results allows further diminishing the number of distributed operations performed, and is popular among industrial systems.

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. Almost all the considered industrial systems (31 out of 33, *i.e.*, 94%) 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[20] 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. In SPARQL federation engines, data masking is provided by allowing hiding named

[20] https://community.denodo.com/kb/view/document/Denodo%20Security%20Overview

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. Subscript $_{ng}$ denotes the use of named graph-based solutions to hide (mask) sensitive information in selected graphs to certain users, and possibly (for AnzoGraph DB) expose sanitized named graph views

System	Data security				
	Authentication	Authorization	Auditing	Encryption	Data masking
AllegroGraph	✓	✓	–	–	–
Amazon Athena	✓	✓	✓	✓	–
Amazon Neptune	✓	✓	✓	✓	–
AnzoGraph DB	✓	✓	–	–	✓$_{ng}$
Apache Drill	✓	✓	–	✓	–
Apache Jena	✓	–	–	–	–
Apache Spark	✓	✓	–	✓	–
BigDAWG	–	–	–	–	–
Blazegraph	–	–	–	–	–
CloudMdsQL	–	–	–	–	–
Comunica	–	–	–	–	–
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	✓	✓	✓	✓	–
Metaphactory	✓	✓	–	–	–
Myria	–	–	–	–	–
Neo4j (Fabric)	✓	✓	–	–	–
Obi-Wan	–	–	–	–	–
Odyssey	–	–	–	–	–
Ontario	–	–	–	–	–
Onto-KIT	–	–	–	–	–
Oracle Big Data SQL	✓	✓	–	–	–
Oracle DB (Spatial & Graph)	✓	✓	–	✓	✓
PolyWeb	–	–	–	–	–
Presto	✓	✓	✓	–	–
Querona Data Virtualization	✓	✓	–	✓	✓
RDFLib	–	–	–	–	–
SAFE	✓	✓	–	–	–
SAGE	–	–	–	–	–
SAP HANA	✓	–	–	–	–
SAS Federation Server	✓	✓	–	✓	✓
SemaGrow	–	–	–	–	–
SPLENDID	–	–	–	–	–
SQL Server (PolyBase)	✓	✓	✓	✓	–
Squerall	–	–	–	–	–
Starburst	✓	✓	✓	✓	–
Stardog	✓	✓	–	–	✓$_{ng}$
Teiid	✓	✓	–	✓	–
TIBCO Data Virtualization	✓	✓	–	✓	–
Trino	✓	✓	–	✓	–
Virtuoso	✓	✓	–	✓	–
Number	**32**	**29**	**10**	**20**	**8**

graphs with sensitive information to certain users in AnzoGraph DB[21] and Stardog;[22] in AnzoGraphDB, this mechanism is complemented by "named views"[23] as a way to define (via SPARQL CONSTRUCT queries) sanitized/masked named graphs to be exposed in place of sensitive ones.
4. Data security has rarely been mentioned in the systems developed by academic and research institutions. Among the 18 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, this dimension comprises the graphical, command line, and application programming interface sub-dimensions. Here, we analyze 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). We cover only *documented* (vs. hidden in the code) interfaces and 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 (31 out of 33, *i.e.*, 94%) provide graphical interfaces, which consist mainly in web consoles or web interfaces, and command line interfaces (all 33 industrial systems), which help users to deploy and manage data federation instances. For example, AllegroGraph provides the AllegroGraph Web View,[24] which is a browser-based graphical interface for exploring, querying, and managing AllegroGraph databases, and Teiid provides users with Teiid Console,[25] a web-based administration and monitoring tool.
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 33, *i.e.*, 70% and 55%) 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 33, *i.e.*, 76%), 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 help users in interfacing with these systems using the most popular programming languages, like C, Go, Java, Python, R, and Ruby.
3. The three systems not associated to any interface in the table are all academic (*Fedra*, *HiBISCuS*, *SAFE*). For these systems, the documentation only covers the experiments conducted and indicates, at most, the script (*Fedra*) or the code entry points (*HiBISCuS*) for reproducing the specific experiments.

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 systems often do not follow well-defined release cycles with proper versioning, *e.g.*, CostFed.[26] In such situations, we leave their versions

[21] https://docs.cambridgesemantics.com/anzograph/v2.3/userdoc/acl.htm#Database
[22] https://docs.stardog.com/operating-stardog/security/named-graph-security
[23] https://docs.cambridgesemantics.com/anzograph/v2.2/userdoc/named-views.htm
[24] https://allegrograph.com/products/agwebview/
[25] https://teiid.github.io/teiid-documents/master/content/admin/Teiid_Console.html
[26] https://github.com/dice-group/CostFed

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	–	✓	✓	–	–	–	✓
Apache Spark	✓	✓	✓	✓	–	–	–
BigDAWG	–	✓	–	–	✓	–	–
Blazegraph	✓	✓	–	–	✓	–	–
CloudMdsQL	–	✓	✓	–	–	–	–
Comunica	✓	✓	–	–	–	–	✓
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	✓	✓	✓	✓	✓	–	–
Metaphactory	✓	✓	–	–	✓	–	✓
Myria	✓	✓	–	–	✓	–	–
Neo4j (Fabric)	✓	✓	✓	–	✓	–	–
Obi-Wan	–	✓	–	–	–	–	✓
Odyssey	–	✓	–	–	–	–	–
Ontario	–	✓	–	–	–	–	–
Onto-KIT	✓	–	–	–	–	–	–
Oracle Big Data SQL	✓	✓	–	–	–	–	–
Oracle DB (Spatial & Graph)	✓	✓	–	–	✓	–	✓
PolyWeb	–	–	–	–	–	–	–
Presto	✓	✓	✓	✓	✓	–	–
Querona Data Virtualization	✓	✓	✓	✓	–	✓	–
RDFLib	–	✓	–	–	–	–	–
SAFE	–	–	–	–	–	–	–
SAGE	✓	✓	–	–	–	–	–
SAP HANA	✓	✓	✓	✓	✓	✓	–
SAS Federation Server	✓	✓	✓	✓	✓	–	–
SemaGrow	✓	✓	–	–	–	–	–
SPLENDID	–	✓	–	–	–	–	–
SQL Server (PolyBase)	✓	✓	✓	✓	–	✓	–
Squerall	✓	✓	–	–	–	–	–
Starburst	✓	✓	✓	✓	✓	–	–
Stardog	✓	✓	–	–	✓	–	✓
Teiid	✓	✓	✓	✓	✓	✓	–
TIBCO Data Virtualization	✓	✓	✓	✓	✓	✓	–
Trino	✓	✓	✓	✓	✓	–	–
Virtuoso	✓	✓	✓	✓	✓	✓	✓
Number	**38**	**45**	**24**	**18**	**27**	**7**	**11**

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, even when accounting for the incomplete information of this sub-dimension (see counts in Table 6). Comparatively less used languages include C/C++ (AnzoGraph DB, SAP HANA and other systems), Python (RDFLib, Squerall, SAGE), Scala (Apache Spark, Ontario), JavaScript (Comunica) and Lisp (AllegroGraph, in combination with Java).
2. Excluding two SaaS industrial systems from Amazon (Athena, Neptune), on-premises deployment is always offered, represents the only available option for academic systems, and concerns software both in native form ($_n$ subscript, almost always possible) and containerized form ($_c$ subscript, *e.g.*, via Docker images), the latter supported more in industrial systems (21 out of 33, *i.e.*, 64%) than academic systems (4 out of 18, *i.e.*, 22%). SaaS (6 industrial systems out of 33, *i.e.*, 18%) is less frequent than IaaS/PaaS (12 out of 33, *i.e.*, 36%), the latter always supporting Amazon AWS ($_a$ subscript), followed by Microsoft Azure ($_m$ subscript, 8 IaaS/PaaS cases out of 12, *i.e.*, 67%) and Google Cloud Platform ($_g$ subscript, 5 IaaS/PaaS cases out of 12, *i.e.*, 42%).
3. Among the industrial systems, the majority are closed source (21 out of 33, *i.e.*, 64%), 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 12, *i.e.*, 58%). Academic systems are all open source without commercial support.
4. 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, as information produced for one system cannot be directly used in other systems.

Other capabilities. Among the other capabilities beyond the data federation task itself (cf. Section 2.3), only data security was captured by our evaluation framework, which is based solely on the aspects of interest arising from applying the methodology of Section 3.2. This fact further remarks the importance of data security, especially among industrial systems, whereas data update and data quality have been less investigated in combination with the data federation. Nevertheless, some of the considered systems provide capabilities related to data update and data quality. Concerning data update over the federated data sources, Teiid[27] and Denodo[28] support `INSERT` and `DELETE` operators, while RDF4J (FedX)[29] supports SPARQL `UPDATE` over the federated SPARQL endpoints. Other systems mention data update, however it is unclear from the systems' documentation whether these updates can be performed on the data sources in the federation, or on the data stored locally by the system itself (*e.g.*, for database systems extended with federation facilities). Concerning data quality, SAS Federation Server[30] supports

[27] https://teiid.github.io/teiid-documents/master/content/reference/as_updatable-views.html

[28] https://community.denodo.com/docs/html/browse/8.0/en/vdp/vql/inserts_updates_and_deletes_over_views/inserts_updates_and_deletes_over_views

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

[30] https://documentation.sas.com/api/docsets/fedsrvag/4.2/content/fedsrvag.pdf

Table 6

Evaluation of *development* dimension. Academic systems in *italics*. "F." and "L." denote "First" and "Latest" respectively. Subscript letters further qualify available deployment options: n = native; c = containerized; a = Amazon AWS; m = Microsoft Azure; g = Google Cloud Platform. "–" denotes feature/information not found in the systems' official documentation, websites, or academic publications, to the best of our efforts

System	Main development language			Deployment			Comm. support	Open source	Release			
	C/C++	Java	Others	On-prem	IaaS/PaaS	SaaS			F. Year	F. Version	L. Year	L. Version
AllegroGraph	–	✓	Lisp	$✓_{nc}$	$✓_{am}$	–	✓	–	2004	6.4.0	2021	7.2.0
Amazon Athena	–	✓	–	–	–	$✓_{a}$	✓	–	2017	–	2021	–
Amazon Neptune	–	✓	–	–	–	$✓_{a}$	✓	–	2018	1.0.1.0	2021	1.0.5.1
AnzoGraph DB	✓	–	–	$✓_{nc}$	$✓_{a}$	–	✓	–	-	2.0	2021	2.3
Apache Drill	–	✓	–	$✓_{nc}$	–	–	–	✓	2012	M1	2021	1.19
Apache Jena	–	✓	–	$✓_{n}$	–	–	–	✓	2012	2.7.0	2021	4.2.0
Apache Spark	–	–	Scala	$✓_{nc}$	–	–	–	✓	2014	1.0	2021	3.2.1
BigDAWG	–	✓	–	$✓_{n}$	–	–	–	✓	2015	–	2017	0.0.5
Blazegraph	–	✓	–	$✓_{n}$	–	–	–	✓	2019	2.1.5	2020	2.1.6rc
CloudMdsQL	–	✓	–	$✓_{n}$	–	–	–	✓	2017	–	2017	–
Comunica	–	–	JavaScript	$✓_{nc}$	–	–	–	✓	2018	1.0.0	2021	1.22.3
CostFed	–	✓	–	$✓_{nc}$	–	–	–	✓	2016	–	2018	–
DARQ	–	✓	–	$✓_{n}$	–	–	–	✓	2006	–	2008	–
Data Virtuality	–	–	–	$✓_{nc}$	–	–	✓	–	–	–	2021	2.4
Denodo	–	–	–	$✓_{nc}$	$✓_{amg}$	–	✓	–	2002	1.0	2020	8.0
Dremio	–	✓	–	$✓_{nc}$	$✓_{am}$	–	✓	✓	2017	1.1	2021	19.0
FEDRA	–	✓	–	$✓_{n}$	–	–	–	✓	2015	–	2015	–
FedX (RDF4J)	–	✓	–	$✓_{n}$	–	–	✓	✓	2011	–	2021	3.7.4
GraphDB	–	✓	–	$✓_{nc}$	–	–	✓	–	2015	6.2	2021	9.10
HiBISCuS	–	✓	–	$✓_{n}$	–	–	–	✓	2014	1	2014	1
IBM Cloud Pak for Data	–	–	–	$✓_{c}$	$✓_{amg}$	✓	✓	–	2018	2.1.0	2021	4.0
IBM Db2 Big SQL	–	✓	–	$✓_{n}$	–	✓	✓	–	2017	–	2020	7.1.0
IBM InfoSphere Federation Server	–	–	–	$✓_{n}$	–	–	✓	–	–	–	2019	10.5.0
JBoss Data Virtualization	–	✓	–	$✓_{nc}$	–	–	✓	✓	2014	6.0.0	2018	6.4.0
Metaphactory	–	–	–	✓	$✓_{a}$	–	–	–	2015	–	2021	4.3.0
Myria	–	✓	–	$✓_{n}$	–	–	–	✓	2014	1	2017	1
Neo4j (Fabric)	–	✓	–	$✓_{nc}$	$✓_{amg}$	✓	✓	✓	2020	4.0.11	2021	4.3.7
Obi-Wan	–	✓	–	$✓_{n}$	–	–	–	✓	2020	–	2020	–
Odyssey	–	✓	–	$✓_{n}$	–	–	–	✓	2016	–	2019	–
Ontario	–	–	Python	$✓_{n}$	–	–	–	✓	2018	–	2021	–
Onto-KIT	–	✓	–	$✓_{n}$	–	–	–	✓	2020	–	2020	–
Oracle Big Data SQL	–	–	–	$✓_{n}$	–	–	–	–	–	3.0.1	2021	4.1.1
Oracle DB (Spatial & Graph)	–	–	–	$✓_{nc}$	–	–	✓	–	2016	–	2021	21c
PolyWeb	–	✓	–	$✓_{n}$	–	–	–	✓	2017	–	2017	–
Presto	–	✓	–	$✓_{nc}$	–	–	✓	✓	2013	0.54	2021	0.265.1
Querona Data Virtualization	–	–	–	$✓_{n}$	–	–	✓	–	2015	–	2020	–
RDFLib	–	–	Python	$✓_{n}$	–	–	–	✓	2002	1.1.1	2021	6.1.1
SAFE	–	✓	–	$✓_{n}$	–	–	–	✓	2017	–	2017	–
SAGE	–	–	Python	$✓_{nc}$	–	–	–	✓	2019	1.1	2021	2.3
SAP HANA	✓	–	–	$✓_{nc}$	$✓_{ag}$	–	✓	–	2018	1.0.SPS12	2020	2.0.SPS05
SAS Federation Server	✓	–	–	$✓_{n}$	–	–	✓	–	2013	3.2	2021	4.4
SemaGrow	–	✓	–	$✓_{nc}$	–	–	–	✓	2014	1.0	2021	2.2.1
SPLENDID	–	✓	–	$✓_{n}$	–	–	–	✓	2011	–	2011	–
SQL Server (PolyBase)	✓	–	–	$✓_{nc}$	–	–	✓	–	2016	2016	2019	2019
Squerall	–	–	Python	$✓_{n}$	–	–	–	✓	2018	0.1	2019	0.2
Starburst	–	✓	–	$✓_{nc}$	$✓_{amg}$	–	✓	–	2019	0.188-e	2021	364-e LTS
Stardog	–	✓	–	$✓_{nc}$	$✓_{a}$	✓	✓	–	2011	0.7.3	2021	7.7.3
Teiid	–	✓	–	$✓_{nc}$	–	–	✓	✓	2009	6.0.0	2020	16.0.0
TIBCO Data Virtualization	–	–	–	$✓_{nc}$	$✓_{am}$	–	✓	–	2007	7.0.5	2021	8.4.0
Trino	–	✓	–	$✓_{nc}$	–	–	✓	✓	–	0.54	2021	364
Virtuoso	✓	–	–	$✓_{nc}$	$✓_{am}$	–	✓	–	–	–	2020	8.3
Number	**5**	**31**	**7**	**49**	**12**	**6**	**26**	**30**	**–**	**–**	**–**	**–**

methods and rules specified in a "SAS Quality Knowledge Base" (QKB), while Stardog[31] supports data quality constraints expressed in SHACL [128]. Given the current steady growth of data scale and variety, we expect these aspects to become increasingly important in the context of data federation.

Ontology-based data access. Ontologies, providing a shared abstraction of a domain of interest, can play a key role in handling the heterogeneity of concepts in data integration. The so-called *Ontology-Based Data Access* (OBDA) approach has been studied intensively [37, 77, 129–134] in the last two decades. In OBDA, a mediating ontology provides a high-level representation of the data contained in a relational source, as well as an encoding of domain knowledge. The link between the ontology and the source is realized through mappings, *e.g.*, expressed using R2RML [135]. The distinctive characteristics of OBDA are that query answers are enriched through *automated reasoning* over the ontology, and that such process is carried out in a *virtual* mode: the data in the database is not materialized as a graph, but rather queries are *rewritten* on-the-fly and executed against the original source.

The virtual characteristic of OBDA makes it a potential candidate for incorporating mediating ontologies in the data federation framework. Still, this marriage has rarely been discussed or considered to its fullest extent, and it represents an open research line. For instance, *Squerall* [96] and *PolyWeb* [32, 85] are virtual systems based on RML/R2RML mappings but both lack reasoning support, hence they do not qualify as fully-fledged OBDA systems as per their definition in the literature [129]. An exception is Obi-Wan [75, 76], an OBDA system[32] able to integrate heterogeneous data sources, including relational, graph-based, and NoSQL ones. Its main idea follows the classical OBDA framework by first rewriting the original queries based on the ontology and the mappings, and then using the mediator system Tatooine [78] to evaluate the rewritten queries over multiple and heterogeneous data sources.

Obi-Wan is for the most part a proof-of-concept of a more general and insightful theoretical exercise. Hence, it does not present any optimization technique specific to the federated setting and is not tailored towards handling real-world, complex scenarios. 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. A preliminary investigation towards this direction has been conducted by Gu et al. [136, 137]. 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.

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 of 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.

Most advanced methods and systems handle overlapping to some extent. BigDAWG exploits equivalence and containment information provided by data curators [138] to identify equivalent operations across different data models, so as to optimize its source-selection strategy. DAW [139], not considered in this survey, exploits a compact representation of data as vectors for which estimates on overlapping can be automatically found. This information is then used to prune, with high recall, redundant sources during source selection. FEDRA [64] does not require to encode data, but relies on *fragment descriptions* for its source selection, where each fragment essentially describe the triples that can be extracted out of a set of data sources.

Note that all approaches require a substantial amount of meta-information which might be hard or even impossible to produce automatically. It has been recently observed by Gu et al. [136, 137] that this limitation is greatly reduced in OBDA settings, where one can exploit both the semantic information provided by the ontology and the URI construction rules encoded in the mappings. This fact allows for optimizations that are not specific to the source

[31] https://docs.stardog.com/data-quality-constraints

[32] Although, based on GLAV mappings as opposed to GAV mappings usually applied in OBDA contexts.

selection phase, such as the removal of redundant or empty operators, or the automatic leveraging of materialization of pre-computed results and on-the-fly access to the sources.

Combining systems. The capabilities of a system can be extended through combination with other tools. We identify two mechanisms for combining a data federation system with a tool, the latter operating as *adapter* and possibly being a data federation system itself; these mechanisms can be iteratively applied to combine multiple components.

In the first mechanism, the tool acts as a source of the system and is used to add *indirect* support for some additional sources that cannot be natively connected to the system, by adapting them to one of the supported source types (e.g., JDBC or ODBC). For instance, the data sources directly supported by Querona Data Virtualization exclude MongoDB but include Denodo and Apache Drill, which instead support MongoDB and can be thus combined to add indirect support for MongoDB. As another example, one may extend SPARQL-based federation to relational sources through the combination with an OBDA engine, as successfully applied by Sima et al. [140] who use the OBDA system Ontop to expose biomedical data as RDF graphs, then federated through a SPARQL federation engine.

In the second mechanism, the tool acts as client of the system and is used to adapt or extend the unified schema, query language(s) or capabilities offered by the system. For example, one may deploy[33] an OBDA engine like Ontop over a SQL-based data federation system such as Dremio or Denodo, so to provide *indirect* support for an RDF/OWL unified schema and SPARQL as unified query language. From a complementary perspective, this combination mechanism can be also seen as adding federation capabilities to the employed tool (the OBDA engine in the example), effectively giving birth to a new data federation system.

As remarked in the text (Section 6.1), the "Data source" dimension of Table 2 and in general all the dimensions and tables of this survey do not account for the combination of systems, but rather focus solely on sources and capabilities that are *directly* supported by the data federation system. The reason is that it is very difficult to comprehensively assess which sources or capabilities a data federation system may acquire by carefully combining it with other tools, as combinations are possibly limitless and the assessment of the practical feasibility of each is non-trivial and not clearly defined, as there might be hard-to-quantify integration costs involved (*e.g.*, to remove minor incompatibilities at the interface between combined tools).

7. Related work

In this survey, we have investigated and analyzed a total of 51 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 review existing approaches, techniques, and systems for both virtual and materialized data integration.

Database community. Halevy et al. [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 by Magnani and Montesi [141] reports on the techniques for managing uncertainty in data integration, and the survey by Bikakis et al. [142] investigates the approaches focusing on semi-structured data. Finally, the works by Arputhamary et al. [143–145] 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 several related surveys [9, 42, 43, 146]. In particular, the survey by Sheth and Larson [9] discusses data federation systems. The authors define terminology and 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). In a different survey, Bondiombouy and Valduriez [146] investigate multistore systems by first introducing the currently available cloud data management and

[33] https://ontop-vkg.org/tutorial/federation/

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 by Tan et al. [43] 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 the work by Sheth and Larson [9], 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. Azevedo et al. [42] focus 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 review 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. Wache, Noy, Ekaputra et al. [147–149] provide general surveys of those solutions for integrating data that are based on Semantic Web technologies and that follow the so-called Ontology-Based Data Integration (OBDI) approach. OBDI is a broader approach than OBDA, and differs from the latter for the fact of allowing for very expressive ontology languages while dropping the requirement of virtual access to data. Hence, OBDI approaches are not really suited to the federation setting considered in this survey. Other works focus instead on specific subdomains in which semantic technologies have been applied to integrate data. In particular, Buccella et al. [150] analyze and compare existing approaches for ontology-driven geographic information integration. An investigation of the approaches and techniques developed in the ontology community for integrating biological data is given by Hassan et al. [151]. The survey by Mountantonakis and Tzitzikas [152] 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 [20, 38–41] surveying, in particular, approaches and systems for federated SPARQL query answering. To summarize, the survey by Rakhmawati et al. [38] 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. Oguz et al. [20] 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". Ngonga Ngomo and Saleem [39] provide an overview of current challenges and opportunities of federated query processing as well as summarize the results of recent state-of-the-art studies. Saleem et al. [40] 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 [153] 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, Qudus et al. [41] first propose 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. Then, they present an empirical evaluation of 5 cost-based SPARQL federation engines on LargeRDFBench [154] according to the proposed metrics.

Comparison. This survey builds on the aforementioned literature and is consistent with the terminology, concepts and key distinctions adopted therein. For instance, considering the foundational work by Sheth and Larson [9], their terminology can be related to several (sub-)dimensions of our evaluation framework as follows: (i) "Heterogeneity" is captured at different levels by our *Data source*, *Query language*, *Federation technique* and various *Development* sub-dimensions; (ii) "Query processing and optimization" is also captured by our *Federation technique*

sub-dimension; (iii) "Access Control" is related to our *Data security* dimension (especially, its *Authorization* sub-dimension). (iv) "Transparency" is captured by the distinction between *Transparent* vs. *Explicit* federation.

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 51 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 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. G. Xiao is supported by the Norwegian Research Council via the SIRIUS Centre for Research Based Innovation (grant No. 237898). 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. We would also like to thank the reviewers for their valuable feedback and comments on earlier versions of this article.

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] H. Betz, F. Gropengießer, K. Hose and K. Sattler, Learning from the History of Distributed Query Processing - A Heretic View on Linked Data Management, 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

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

[24] 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/.

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

[26] 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.

[27] 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.

[28] 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.

[29] 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.
[30] 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.
[31] 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.
[32] 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.
[33] 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.
[34] 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.
[35] 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.
[36] 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.
[37] 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.
[38] 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.
[39] 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.
[40] 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.
[41] 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.
[42] 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.
[43] 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.
[44] Apache Drill, Accessed 18 November 2021. https://drill.apache.org/.
[45] 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.
[46] 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.
[47] AllegroGraph, Accessed 18 November 2021. https://allegrograph.com/.
[48] Amazon Athena, Accessed 18 November 2021. https://docs.aws.amazon.com/athena/latest/ug/work-with-data-stores.html.
[49] Presto, Accessed 18 November 2021. https://prestodb.io/.
[50] Amazon Neptune, Accessed 18 November 2021. https://aws.amazon.com/neptune/.
[51] Anzograph, Accessed 17 November 2021. https://www.cambridgesemantics.com/anzograph/.
[52] 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.
[53] 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.
[54] Jena, Accessed 18 November 2021. https://jena.apache.org/documentation/query/.
[55] Spark SQL, Accessed 18 November 2021. https://spark.apache.org/sql/.
[56] 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.
[57] 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.
[58] Blazegraph, Accessed 6 December 2021. https://blazegraph.com/.
[59] 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.
[60] 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.

[61] 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.
[62] Denodo, Accessed 17 Novemebr 2021. https://www.denodo.com/en.
[63] Dremio, Accessed 17 November 2021. https://www.dremio.com/.
[64] 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.
[65] GraphDB, Accessed 17 November 2021. https://graphdb.ontotext.com/.
[66] 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.
[67] IBM Cloud Pak for Data, Accessed 17 November 2021. https://www.ibm.com/products/cloud-pak-for-data.
[68] IBM Db2 Big SQL, Accessed 18 November 2021. https://www.ibm.com/products/db2-big-sql.
[69] IBM InfoSphere Federation Server, Accessed 18 November 2021. https://www.ibm.com/docs/en/iis/11.7?topic=components-infosphere-federation-server.
[70] JBoss Data Virtualization, Accessed 17 November 2021. https://developers.redhat.com/products/datavirt/overview.
[71] Metaphactory, Accessed 18 November 2021. https://metaphacts.com/product.
[72] 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.
[73] Neo4j, Accessed 17 November 2021. https://neo4j.com/.
[74] 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.
[75] 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.
[76] 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.
[77] 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.
[78] 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.
[79] 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.
[80] 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.
[81] 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.
[82] Oracle Big Data SQL, Accessed 18 November 2021. https://www.oracle.com/database/technologies/datawarehouse-bigdata/bigdata-sql.html.
[83] Oracle Spatial and Graph, Accessed 16 November 2021. https://www.oracle.com/database/technologies/spatialandgraph.html.
[84] 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.
[85] 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.
[86] 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.
[87] Querona Data Virtualization, Accessed 17 November 2021. https://www.querona.io/.
[88] RDFLib, Accessed 26 July 2022. https://rdflib.readthedocs.io/en/stable/.
[89] 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.
[90] T. Minier, H. Skaf-Molli and P. Molli, SaGe: Web Preemption for Public SPARQL Query Services, in: *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, L. Liu, R.W. White, A. Mantrach, F. Silvestri, J.J. McAuley, R. Baeza-Yates and L. Zia, eds, ACM, 2019, pp. 1268–1278. doi:10.1145/3308558.3313652.
[91] SAP HANA (Smart Data Access), Accessed 18 November 2021. https://help.sap.com/docs/SAP_HANA_PLATFORM/6b94445c94ae495c83a19646e7c3fd56/a07c7ff25997460bbcb73099fb59007d.html?locale=en-US&version=2.0.05.
[92] SAS Federation Server, Accessed 18 November 2021. https://support.sas.com/en/software/federation-server-support.html.
[93] 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.
[94] 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.

[95] SQL Server (PolyBase), Accessed 18 November 2021. https://docs.microsoft.com/en-us/sql/relational-databases/polybase/polybase-guide?view=sql-server-ver15.
[96] 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.
[97] Starburst, Accessed 18 December 2021. https://www.starburst.io/.
[98] Stardog, Accessed 17 November 2021. https://www.stardog.com/.
[99] TIBCO Data Virtualization, Accessed 17 November 2021. https://www.tibco.com/products/data-virtualization.
[100] Trino, Accessed 18 November 2021. https://trino.io/.
[101] Virtuoso, Accessed 17 November 2021. https://virtuoso.openlinksw.com/.
[102] O. Erling, Virtuoso, a Hybrid RDBMS/Graph Column Store, *IEEE Data Engineering Bull.* **35**(1) (2012), 3–8.
[103] 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.
[104] P.J. Sadalage and M. Fowler, *NoSQL Distilled: a Brief Guide to the Emerging World of Polyglot Persistence*, Pearson Education, 2013.
[105] DB-Engines, Accessed 16 February 2022. https://db-engines.com/en/.
[106] Database of Databases, Accessed 16 February 2022. https://dbdb.io/.
[107] S. Konstantopoulos, A. Charalambidis, A. Troumpoukis, G. Mouchakis and V. Karkaletsis, The Sevod Vocabulary for Dataset Descriptions for Federated Querying, 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.
[108] H. Stuckenschmidt, R. Vdovjak, G.-J. Houben and J. Broekstra, Index Structures and Algorithms for Querying Distributed RDF Repositories, in: *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, Association for Computing Machinery, New York, NY, USA, 2004, pp. 631–639–. ISBN 158113844X. doi:10.1145/988672.988758.
[109] M.T. Özsu and P. Valduriez, *Principles of Distributed Database Systems, 4th Edition*, Springer, 2020. ISBN 978-3-030-26252-5. doi:10.1007/978-3-030-26253-2.
[110] L. Haas, D. Kossmann, E. Wimmers and J. Yang, Optimizing Queries across Diverse Data Sources, in: *23rd International Conference on Very Large Data Bases (VLDB 1997)*, 1997.
[111] Data Virtuality (Push down), Accessed 17 November 2021. https://documentation.datavirtuality.com/24/reference-guide/federated-planning/federated-optimizations#FederatedOptimizations-Pushdown.
[112] IBM Db2 Big SQL: What's new and changed in Data Virtualization, Accessed 18 November 2021. https://www.ibm.com/docs/en/cloud-paks/cp-data/4.5.x?topic=new-data-virtualization.
[113] SQL Server (PolyBase Push down), Accessed 18 November 2021. https://docs.microsoft.com/en-us/sql/relational-databases/polybase/polybase-pushdown-computation?view=sql-server-ver15.
[114] Starburst Push down, Accessed 18 December 2021. https://docs.starburst.io/latest/optimizer/pushdown.html.
[115] Trino (Pushdown), Accessed 18 November 2021. https://trino.io/docs/current/optimizer/pushdown.html.
[116] A. Silberschatz, H.F. Korth and S. Sudarshan, *Database System Concepts, Seventh Edition*, McGraw-Hill Book Company, 2020. ISBN 9780078022159.
[117] Teiid (Planning Overview), Accessed 16 November 2021. http://teiid.github.io/teiid-documents/16.0.x/content/reference/r_planning-overview.html.
[118] SAP HANA (SQL Optimizer), Accessed 18 November 2021. https://help.sap.com/docs/SAP_HANA_PLATFORM/9de0171a6027400bb3b9bee385222eff/d2948cc2209a407ea2b686c29e72ca50.html.
[119] S. Blanas, J.M. Patel, V. Ercegovac, J. Rao, E.J. Shekita and Y. Tian, A comparison of join algorithms for log processing in MaPreduce, in: *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2010, Indianapolis, Indiana, USA, June 6-10, 2010*, ACM, 2010, pp. 975–986. doi:10.1145/1807167.1807273.
[120] Apache Drill (Broadcast Join), Accessed 18 November 2021. https://drill.apache.org/docs/join-planning-guidelines/.
[121] Starburst (Broadcast Join), Accessed 18 December 2021. https://docs.starburst.io/latest/admin/properties-general.html.
[122] A.M. Gupta, V. Gadepally and M. Stonebraker, Cross-engine query execution in federated database systems, 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.2016.7761648.
[123] Presto (Choosing Presto Join and Sort Algorithms), Accessed 18 November 2021. https://docs.treasuredata.com/display/public/PD/Choosing+Presto+Join+and+Sort+Algorithms.
[124] Trino, Accessed 18 November 2021. https://trino.io/docs/current/optimizer/cost-based-optimizations.html#join-distribution-selection.
[125] Data Federation (Data Movement), Accessed 12 March 2021. https://community.denodo.com/docs/html/browse/8.0/vdp/administration/optimizing_queries/data_movement/data_movement.
[126] 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.
[127] ISO/IEC JTC 1/SC 32 Data management and interchange, ISO/IEC 9075-9:2016 – Information technology – Database languages – SQL – Part 9: Management of External Data (SQL/MED), Technical Report, ISO/IEC, 2016.
[128] D. Kontokostas and H. Knublauch, Shapes Constraint Language (SHACL), W3C Recommendation, W3C, 2017. https://www.w3.org/TR/2017/REC-shacl-20170720/.

[129] 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.
[130] 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.
[131] 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.
[132] 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.
[133] 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.
[134] 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.
[135] 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/.
[136] Z. Gu, D. Lanti, A. Mosca, G. Xiao, J. Xiong and D. Calvanese, Ontology-Based Data Federation, in: *Proc. of the 35th Int. Workshop on Description Logics (DL)*, CEUR Workshop Proceedings, 2022, To appear in proceedings.
[137] Z. Gu, D. Lanti, A. Mosca, G. Xiao, J. Xiong and D. Calvanese, Ontology-based Data Federation, in: *The 11th International Joint Conference on Knowledge Graphs (IJCKG 2022)*, 2022, To appear in print.
[138] Z. She, S. Ravishankar and J. Duggan, BigDAWG polystore query optimization through semantic equivalences, 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.2016.7761584.
[139] 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.
[140] A.C. Sima, T.M. de Farias, E. Zbinden, M. Anisimova, M. Gil, H. Stockinger, K. Stockinger, M. Robinson-Rechavi and C. Dessimoz, Enabling semantic queries across federated bioinformatics databases, *Database J. Biol. Databases Curation* **2019** (2019), baz106. doi:10.1093/database/baz106.
[141] 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.
[142] 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.
[143] B. Arputhamary and L. Arockiam, A review on Big Data Integration, *Int. J. Computer Applications* **22**(3) (2015), 21–26.
[144] X.L. Dong and D. Srivastava, *Big Data Integration*, Synthesis Lectures on Data Management, Morgan & Claypool Publishers, 2015. doi:10.2200/S00578ED1V01Y201404DTM040.
[145] 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.
[146] 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.
[147] 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.
[148] N.F. Noy, Semantic Integration: A Survey Of Ontology-Based Approaches, *SIGMOD Record* **33**(4) (2004), 65–70. doi:10.1145/1041410.1041421.
[149] 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.
[150] 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.
[151] 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.
[152] 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.
[153] 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.
[154] 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.
[155] 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.

[156] N.A. Rakhmawati, An Holistic Evaluation of Federated SPARQL Query Engine, in: *Proc. of Information Systems International Conference (ISICO)*, 2013.
[157] 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.
[158] 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.
[159] O. Görlitz, Distributed query processing for federated RDF data management, PhD thesis, University of Koblenz-Landau, 2015. http://kola.opus.hbz-nrw.de/volltexte/2015/1091/pdf/diss_print_final.pdf.
[160] 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.
[161] 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.
[162] 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.
[163] 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.
[164] 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.
[165] H. Stuckenschmidt, R. Vdovjak, J. Broekstra and G. Houben, Towards distributed processing of RDF path queries, *Int. J. Web Eng. Technol.* **2**(2/3) (2005), 207–230. doi:10.1504/IJWET.2005.008484.
[166] J. Zemánek and S. Schenk, Optimizing SPARQL Queries over Disparate RDF Data Sources through Distributed Semi-Joins, in: *Proceedings of the Poster and Demonstration Session at the 7th International Semantic Web Conference (ISWC2008), Karlsruhe, Germany, October 28, 2008*, CEUR Workshop Proceedings, Vol. 401, CEUR-WS.org, 2008.
[167] S. Schenk and S. Staab, Networked graphs: a declarative mechanism for SPARQL rules, SPARQL views and RDF data integration on the web, in: *Proceedings of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21-25, 2008*, ACM, 2008, pp. 585–594. doi:10.1145/1367497.1367577.
[168] 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.
[169] 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.
[170] Z. Kaoudi, M. Koubarakis, K. Kyzirakos, I. Miliaraki, M. Magiridou and A. Papadakis-Pesaresi, Atlas: Storing, updating and querying RDF(S) data on top of DHTs, *J. Web Semant.* **8**(4) (2010), 271–277. doi:10.1016/j.websem.2010.07.001.
[171] A. Harth, K. Hose, M. Karnstedt, A. Polleres, K. Sattler and J. Umbrich, Data summaries for on-demand queries over linked data, in: *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, M. Rappa, P. Jones, J. Freire and S. Chakrabarti, eds, ACM, 2010, pp. 411–420. doi:10.1145/1772690.1772733.
[172] G. Ladwig and T. Tran, Linked Data Query Processing Strategies, in: *The Semantic Web - ISWC 2010 - 9th International Semantic Web Conference, ISWC 2010, Shanghai, China, November 7-11, 2010, Revised Selected Papers, Part I*, Lecture Notes in Computer Science, Vol. 6496, Springer, 2010, pp. 453–469. doi:10.1007/978-3-642-17746-0_29.
[173] M. Acosta, M.-E. Vidal, T. Lampo, J. Castillo and E. Ruckhaus, ANAPSID: An Adaptive Query Processing Engine for SPARQL Endpoints, in: *Proc. of Int. Semantic Web Conference*, LNCS, Vol. 7031, Springer, 2011, pp. 18–34. doi:10.1007/978-3-642-25073-6_2.
[174] S. Lynden, I. Kojima, A. Matono and Y. Tanimura, Aderis: An adaptive query processor for joining federated sparql endpoints, in: *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, 2011, pp. 808–817.
[175] X. Wang, T. Tiropanis and H.C. Davis, Querying the Web of Data with Graph Theory-based Techniques, Technical Report, University of Southampton, 2011.
[176] X. Wang, T. Tiropanis and H.C. Davis, Evaluating Graph Traversal Algorithms for Distributed SPARQL Query Optimization, in: *The Semantic Web - Joint International Semantic Technology Conference, JIST 2011, Hangzhou, China, December 4-7, 2011. Proceedings*, J.Z. Pan, H. Chen, H. Kim, J. Li, Z. Wu, I. Horrocks, R. Mizoguchi and Z. Wu, eds, Lecture Notes in Computer Science, Vol. 7185, Springer, 2011, pp. 210–225. doi:10.1007/978-3-642-29923-0_14.
[177] G. Ladwig and T. Tran, SIHJoin: Querying Remote and Local Linked Data, in: *The Semantic Web: Research and Applications - 8th Extended Semantic Web Conference, ESWC 2011, Heraklion, Crete, Greece, May 29-June 2, 2011, Proceedings, Part I*, Lecture Notes in Computer Science, Vol. 6643, Springer, 2011, pp. 139–153. doi:10.1007/978-3-642-21034-1_10.
[178] 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.

[179] F. Prasser, A. Kemper and K.A. Kuhn, Efficient distributed query processing for autonomous RDF databases, in: *15th International Conference on Extending Database Technology, EDBT '12, Berlin, Germany, March 27-30, 2012, Proceedings*, ACM, 2012, pp. 372–383. doi:10.1145/2247596.2247640.

[180] O. Mora, G. Engelbrecht and J. Bisbal, A Service-Oriented Distributed Semantic Mediator: Integrating Multiscale Biomedical Information, *IEEE Trans. Inf. Technol. Biomed.* **16**(6) (2012), 1296–1303. doi:10.1109/TITB.2012.2215045.

[181] 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.

[182] A. Hasnain, M.R. Kamdar, P. Hasapis, D. Zeginis, C.N.W. Jr., H.F. Deus, D. Ntalaperas, K.A. Tarabanis, M. Mehdi and S. Decker, Linked Biomedical Dataspace: Lessons Learned Integrating Data for Drug Discovery, in: *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I*, P. Mika, T. Tudorache, A. Bernstein, C. Welty, C.A. Knoblock, D. Vrandecic, P. Groth, N.F. Noy, K. Janowicz and C.A. Goble, eds, Lecture Notes in Computer Science, Vol. 8796, Springer, 2014, pp. 114–130. doi:10.1007/978-3-319-11964-9_8.

[183] A. Hasnain, S.S. e Zainab, M.R. Kamdar, Q. Mehmood, C.N.W. Jr., Q.A. Fatimah, H.F. Deus, M. Mehdi and S. Decker, A Roadmap for Navigating the Life Sciences Linked Open Data Cloud, in: *Semantic Technology - 4th Joint International Conference, JIST 2014, Chiang Mai, Thailand, November 9-11, 2014. Revised Selected Papers*, T. Supnithi, T. Yamaguchi, J.Z. Pan, V. Wuwongse and M. Buranarach, eds, Lecture Notes in Computer Science, Vol. 8943, Springer, 2014, pp. 97–112. doi:10.1007/978-3-319-15615-6_8.

[184] A. Hasnain, R. Fox, S. Decker and H.F. Deus, Cataloguing and Linking Life Sciences LOD Cloud, in: *Proc. of 1st Int. Workshop on Ontology Engineering in a Data-driven World (OEDW), co-located with EKAW*, 2012.

[185] X. Wang, T. Tiropanis and H.C. Davis, LHD: Optimising Linked Data Query Processing Using Parallelisation, in: *Proceedings of the WWW2013 Workshop on Linked Data on the Web, Rio de Janeiro, Brazil, 14 May, 2013*, CEUR Workshop Proceedings, Vol. 996, CEUR-WS.org, 2013.

[186] 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.

[187] O. Hartig, SQUIN: a traversal based query execution system for the web of linked data, in: *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2013, New York, NY, USA, June 22-27, 2013*, ACM, 2013, pp. 1081–1084. doi:10.1145/2463676.2465231.

[188] 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.

[189] 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.

[190] 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.

[191] 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.

[192] A.J. Elmore, J. Duggan, M. Stonebraker, M. Balazinska, U. Çetintemel, V. Gadepally, J. Heer, B. Howe, J. Kepner, T. Kraska, S. Madden, D. Maier, T.G. Mattson, S. Papadopoulos, J. Parkhurst, N. Tatbul, M. Vartak and S. Zdonik, A Demonstration of the BigDAWG Polystore System, *Proc. VLDB Endow.* **8**(12) (2015), 1908–1911. doi:10.14778/2824032.2824098.

[193] 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.

[194] 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.

[195] 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 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.

[196] D. Collarana, C. Lange and S. Auer, FuhSen: A Platform for Federated, RDF-based Hybrid Search, in: *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11-15, 2016, Companion Volume*, ACM, 2016, pp. 171–174. doi:10.1145/2872518.2890535.

[197] Y. Khan, M. Saleem, A. Iqbal, M. Mehdi, A. Hogan, A.N. Ngomo, S. Decker and R. Sahay, SAFE: Policy Aware SPARQL Query Federation Over RDF Data Cubes, in: *Proceedings of the 7th International Workshop on Semantic Web Applications and Tools for Life Sciences, Berlin, Germany, December 9-11, 2014*, CEUR Workshop Proceedings, Vol. 1320, CEUR-WS.org, 2014.

[198] 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.

[199] E. Mansour, I. Abdelaziz, M. Ouzzani, A. Aboulnaga and P. Kalnis, A Demonstration of Lusail: Querying Linked Data at Scale, in: *Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD Conference 2017, Chicago, IL, USA, May 14-19, 2017*, ACM, 2017, pp. 1603–1606. doi:10.1145/3035918.3058731.

[200] I. Abdelaziz, E. Mansour, M. Ouzzani, A. Aboulnaga and P. Kalnis, Query Optimizations over Decentralized RDF Graphs, in: *33rd IEEE International Conference on Data Engineering, ICDE 2017, San Diego, CA, USA, April 19-22, 2017*, IEEE Computer Society, 2017, pp. 139–142. doi:10.1109/ICDE.2017.59.

[201] 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 (Demonstrations), Snowbird, UT, USA, June 22-27, 2014*, ACM, 2014, pp. 881–884. doi:10.1145/2588555.2594530.

[202] 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. Biomedical Semantics* **8**(1) (2017), 13:1–13:19. doi:10.1186/s13326-017-0118-0.

[203] R. Verborgh, M.V. Sande, O. Hartig, J.V. Herwegen, L.D. Vocht, B.D. Meester, G. Haesendonck and P. Colpaert, Triple Pattern Fragments: A low-cost knowledge graph interface for the Web, *J. Web Semant.* **37-38** (2016), 184–206. doi:10.1016/j.websem.2016.03.003.

[204] 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.

[205] K.M. Endris, Z. Almhithawi, I. Lytra, M. Vidal and S. Auer, BOUNCER: Privacy-Aware Query Processing over Federations of RDF Datasets, in: *Database and Expert Systems Applications - 29th International Conference, DEXA 2018, Regensburg, Germany, September 3-6, 2018, Proceedings, Part I*, Lecture Notes in Computer Science, Vol. 11029, Springer, 2018, pp. 69–84. doi:10.1007/978-3-319-98809-2_5.

[206] 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.

[207] 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.

[208] 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.

[209] Q. Ge, P. Peng, Z. Xu, L. Zou and Z. Qin, FMQO: A Federated RDF System Supporting Multi-query Optimization, in: *Web and Big Data - Third International Joint Conference, APWeb-WAIM 2019, Chengdu, China, August 1-3, 2019, Proceedings, Part II*, Lecture Notes in Computer Science, Vol. 11642, Springer, 2019, pp. 397–401. doi:10.1007/978-3-030-26075-0_30.

[210] 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.

[211] 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.

[212] 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.

[213] 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.

[214] 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.

[215] K.S. Aggour, V.S. Kumar, P. Cuddihy, J.W. Williams, V. Gupta, L. Dial, T. Hanlon, J. Gambone and J. Vinciquerra, Federated Multimodal Big Data Storage & Analytics Platform for Additive Manufacturing, in: *2019 IEEE International Conference on Big Data (IEEE BigData), Los Angeles, CA, USA, December 9-12, 2019*, C.K. Baru, J. Huan, L. Khan, X. Hu, R. Ak, Y. Tian, R.S. Barga, C. Zaniolo, K. Lee and Y.F. Ye, eds, IEEE, 2019, pp. 1729–1738. doi:10.1109/BigData47090.2019.9006495.

[216] 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.

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 on a *local*, *per-system* basis, 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[34] and Progress[35] 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 different from the ones listed in Table 7 (*e.g.*, GraphQL sources via specific connectors[36]) and 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 stored structured files 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 a 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. 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				

[34] https://www.cdata.com/drivers/
[35] https://www.progress.com/connectors
[36] https://www.cdata.com/drivers/graphql/

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	SPARQLp		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		Jena API, SPARQLp				
Apache Spark	Hive, JDBC			any file (content field + metadata), Avro, CSV, JSON, ORC, Parquet		
BigDAWG	PostgreSQL		Accumulo$_w$			SciDB
Blazegraph		SPARQLp				
CloudMdsQL	Derby	Sparksee$_g$	MongoDB$_d$			
Comunica		SPARQLp, TPF		RDF		
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$_a$
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$_*$	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
RDFLib		SPARQLp				
SAFE		SPARQLp				
SAGE		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	SPARQLp				

Appendix B. Selection of academic systems

We report further details about the academic systems selection process described in Section 3.1, providing: (i) the statistics of the 295 academic publications found in our literature search (Section B.1); (ii) the metadata and the considered systems and aspects for the 17 *system comparison* publications found among them (Section B.2); (iii) the inclusion criteria satisfied or violated by the 56 academic systems found, which support our selection of 18 academic systems (Section B.3); and (iv) the full bibliography of all the 295 collected academic publications (Section B.4).

B.1. Statistics of collected publications

Table 8 reports the breakdown of the 295 academic publications collected in our literature review, grouped by year and venue. We distinguish between *journals*, *conferences*, *workshops* and *others* venue categories, the latter comprising PhD thesis, poster and demo papers, technical reports, book chapters and so on. We also highlight the more frequent venues for each category, such as ISWC for conferences, to provide some insight about where the data federation topics of this survey have been mostly discussed. The resulting venues include the main journals and conferences of the Semantic Web and Database areas, as well as some of their co-located workshops.

Figure 6 provides a graphical depiction of the information in Table 8. The pie chart on the left shows that most of the publications found are from conference proceedings. Instead, the stacked area chart on the right suggests an overall increasing trend of yearly publications on data federation in the representative period 2011–2019. Note that before 2011 we do not have enough data so we grouped all years together, while after 2019 counts are not informative as the bulk of our literature review was conducted between the end of 2020 and the beginning of 2021, so we missed some late 2020 publications and we collected only a very small sample of 2021 publications (*e.g.*, articles from "online first" venues that were already online when we conducted our literature search).

B.2. System comparison publications

Table 9 lists the 17 *system comparison publications* contained in the retrieved 295 academic publications. They include surveys, benchmarks, evaluation papers and PhD theses focusing on data federation topics and involving extensive qualitative and/or quantitative comparison of multiple data federation systems. For each system comparison publication, we report in Table 9: (i) its title, venue and year metadata; (ii) the number of citations from Google Scholar (as of 2022/06/07); (iii) the investigated systems, this information being used in our *relevance* criterion for system selection (Section 3.1); and (iv) the aspects of interest analyzed in the publication, this information being exploited in our methodology for the design of the system evaluation framework (Section 3.2). Note that for certain publications, some aspects (marked with *) are investigated only for a subset of the considered systems (also marked with *). Additionally, two surveys [38, 40] can be also found in revised form but with the same systems and aspects considered, in the PhD thesis of the respective authors (this is indicated in the *Title* column of Table 9).

B.3. Academic systems

Table 10 lists all the 56 academic systems identifed starting from the 295 academic publications, with the names of selected systems highlighted in bold. Systems are sorted by year, which we conventionally set as the publication year of the most recent conference or journal paper about the system. Besides the system name, Table 10 contains all the information considered for selecting or discarding a system based on the inclusion criteria of Section 3.1:

- column *C.A.* denotes code availability, which is a mandatory requirement for a system being selected;
- columns *Sur*, *Het*, *Sec* denote the system respectively being mentioned by a survey of Table 9 (in subscript the number of mentioning surveys), supporting heterogeneous data sources, or considering data security; at least one of these features must be present for a system being selected;
- column *# Cit.* denotes the number of citations from Google Scholar (as of 2022/06/07), summed over all the publications about the system; this number must be the largest one for periods $\leqslant$2008, 2009–2011, and 2012–2014, or above the threshold of 10 for the period 2015–2019 (no constraint set for period $\geqslant$2020);
- column *Publication*, finally, lists the academic publications about the system, for which we require *formal* (*i.e.*, peer-reviewed) publications for a system being selected.

Table 8
Statistics of collected publications

	Venue	⩽2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	Total
Journals	J. Web Semant.	1	–	–	1	1	–	1	2	2	–	–	–	**8**
	Semantic Web	–	–	–	–	–	1	1	–	–	1	–	2	**5**
	ACM Comput. Surv.	1	–	–	–	–	–	–	–	1	2	–	–	**4**
	IEEE Trans. Knowl. Data Eng.	–	–	–	–	–	–	–	–	–	–	–	3	**3**
	J. Biomed. Semant.	–	–	–	–	1	–	–	2	–	–	–	–	**3**
	VLDB J.	–	–	–	–	–	1	1	–	–	–	1	–	**3**
	Other journals	9	4	1	3	4	4	8	4	5	6	6	3	**57**
	Total journals	***11***	***4***	***1***	***4***	***6***	***6***	***11***	***8***	***8***	***9***	***7***	***8***	***83***
Conferences	ISWC	3	3	3	3	2	2	1	2	1	1	1	1	**23**
	ESWC	3	3	2	1	1	2	–	2	1	3	1	1	**20**
	WWW	2	–	–	–	–	1	1	–	2	3	–	–	**9**
	VLDB	1	–	–	–	–	–	3	1	–	1	2	–	**8**
	SIGMOD	–	–	–	2	1	1	1	–	1	1	–	–	**7**
	DEXA	–	–	–	–	–	–	–	1	1	3	–	–	**5**
	IEEE BigData	–	–	–	–	–	–	1	2	–	2	–	–	**5**
	iiWAS	–	–	–	1	1	–	–	–	–	1	2	–	**5**
	SEMANTiCS	1	–	–	–	–	1	–	2	1	–	–	–	**5**
	ICDE	1	–	1	–	1	–	–	1	–	1	–	–	**5**
	CIDR	–	–	–	–	–	1	–	1	–	–	1	–	**3**
	EDBT	–	–	2	–	–	–	–	–	–	–	1	–	**3**
	Other conferences	3	2	3	3	3	2	4	4	9	7	4	1	**45**
	Total conferences	***14***	***8***	***11***	***10***	***9***	***10***	***11***	***16***	***16***	***23***	***12***	***3***	***143***
Workshops	PROFILES @ ISWC/ESWC	–	–	–	–	–	–	2	2	–	–	–	–	**4**
	COLD @ ISWC	–	2	1	–	–	–	–	–	–	–	–	–	**3**
	LDOW @ WWW	–	–	1	1	–	–	–	–	1	–	–	–	**3**
	QuWeDa @ ESWC/ISWC	–	–	–	–	–	–	–	1	1	1	–	–	**3**
	AMW	–	–	–	–	1	1	–	–	–	–	–	–	**2**
	Other workshops	2	1	2	3	–	1	–	1	–	1	3	–	**14**
	Total workshops	***2***	***3***	***4***	***4***	***1***	***2***	***2***	***4***	***2***	***2***	***3***	***–***	***29***
Others	PhD theses	–	–	–	–	–	2	–	2	1	2	7	1	**15**
	Posters & demos	2	1	–	1	1	2	1	3	–	1	1	–	**13**
	Other (e.g., books, chapters)	–	2	1	–	1	–	–	–	2	–	4	2	**12**
	Total other	***2***	***3***	***1***	***1***	***2***	***4***	***1***	***5***	***3***	***3***	***12***	***3***	***40***
	Total all venues	**29**	**18**	**17**	**19**	**18**	**22**	**25**	**33**	**29**	**37**	**34**	**14**	**295**

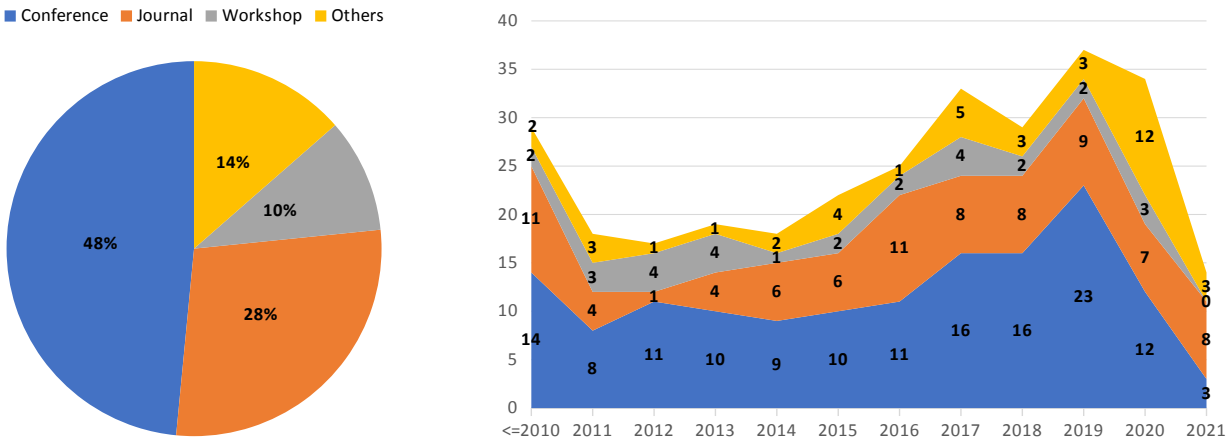

Fig. 6. Number of publications by venue type, both overall (left pie chart) and by year (right stacked area chart).

Table 9

Collected system comparison publications. *Cit.* = citations (2022/06/07); *= restriction to selected aspects/systems; **bold** = systems in our survey

Title	Venues	Year	Cit.	Considered systems	Assessed dimensions
FedBench: A benchmark suite for federated semantic data query processing [153]	ISWC	2011	194	**SPLENDID**, ~~Sesame AliBaba~~	evaluation time, # endpoint requests
A comparison of federation over SPARQL endpoints frameworks [38] *also available as PhD thesis chapter [155]*	KESW	2013	21	ADERIS, ANAPSID, Avalanche, **DARQ**, Distributed SPARQL, **FedX**, GDS, **Jena**, SemWIQ, ~~Sesame AliBaba~~, SPARQL-DQP, **SPLENDID**, **Virtuoso**, WoDQA	catalog, platform, source selection, cache, query execution (join types), source tracking, GUI
An holistic evaluation of federated SPARQL query engine [156]	ISICO	2013	2	**DARQ**, **FedX**, **SPLENDID**	response time, amount of data sent/received, size of intermediate results, # requests, # ASK requests, # sources selected, avg. size of intermediate results, max rows retrieved, requests workload, avg. data received
QFed: Query set for federated SPARQL query benchmark [157]	iiWAS	2014	13	**FedX**, **Jena**	data transmission, run time
On metrics for measuring fragmentation of federation over SPARQL Endpoints [158]	WEBIST	2014	7	**DARQ**, **SPLENDID**	data transfer, requests workload
Distributed query processing for federated RDF data [159]	PhD thesis Univ. Koblenz-Landau	2015	3	ANAPSID, Avalanche, **DARQ**, FeDeRate, **FedX**, Mediator SAIL, Min-Tree BGP, MisMed, Networked Graphs, Prasser at al., QTree, SemWIQ, SPARQL-DQP, **SPLENDID**	catalog, source selection, query optimization (heuristics or cost-based), query execution (join types)
Federated query processing on Linked Data: A qualitative survey and open challenges [20]	Knowl. Eng. Rev.	2015	32	ADERIS, ANAPSID, **DARQ**, **FedX**, LHD, **SPLENDID**, WoDQA	data source selection, join methods, query optimization
A fine-grained evaluation of SPARQL endpoint federation systems [40] *also available as PhD thesis chapter [160, 161]*	Semantic Web	2016	125	ADERIS*, ANAPSID*, Atlas, Avalanche, **DARQ***, DAW, FedSearch, **FedX***, GRANATUM, LDQPS, LHD*, SIHJoin, **SPLENDID***, WoDQA,	category, code availability, implementation language, license, source selection type, join type, cache, index/catalog update, system's features (results completeness and duplicate detection), supported SPARQL constructors # sources selected*, # ASK requests*, result completeness*, source selection time*, query execution time*, overall performance*, effect of data partitioning*
LargeRDFBench: A billion triples benchmark for SPARQL endpoint federation [154]	J. Web Semant.	2018	57	ANAPSID, **FedX**, **HiBISCuS**, **SPLENDID**	source selection, result set completeness and correctness, query execution time
Extending LargeRDFBench for multi-source data at scale for SPARQL endpoint federation [162]	SSWS @ ISWC	2018	4	ANAPSID, **CostFed**, **FedX**, **HiBISCuS**, **SemaGrow**, **SPLENDID**	# source selection (TPWSS), # ASK requests, source selection time, runtime
Enabling query processing across heterogeneous data models: A survey [43]	BigData	2017	101	**Apache Drill**, **BigDAWG**, **CloudMdsQL**, **Myria**	heterogeneity, autonomy, transparency, flexibility, optimality
An empirical evaluation of cost-based federated SPARQL query processing Engines [41]	Semantic Web	2019	1	ANAPSID, BioFed, **CostFed***, **DARQ**, **FedX**, LHD*, Lusail, MULDER, **Odyssey***, **SemaGrow***, **SPLENDID***	index, query processing, network (i.e., # transferred tuples), result set, resources, errors/q-error of triple patterns*, errors/q-error of joins between triple patterns*, errors/q-error of overall query plans*, overall query runtime*, # tuples transferred*, source selection metrics*, quality of generated plans*
Large scale semantic integration of Linked Data: A survey [152]	ACM Comput. Surv.	2019	54	ANAPSID, **DARQ**, DAW, **FedX**, **HiBISCuS**, MULDER, **SPLENDID**	data set types, output types, transformations, schema matching, instance matching, provenance levels, quality, evolution, tested data sets and scalability
Federated query processing over heterogeneous data sources in a semantic data lake [163]	PhD thesis Univ. Bonn	2020	3	ANAPSID, Avalance, **DARQ**, DAW, **FEDRA**, **FedX**, **HiBISCuS**, Lusail, **Odyssey**, **SAFE**, **Semagrow**, **SPLENDID**	catalog, ASK-based source selection, privacy awareness
Modern federated database systems: An overview [42]	ICEIS	2020	3	**Apache Drill**, **BigDAWG**, **CloudMdsQL**, **Myria**	definition, owner, goal, internal data representation and platform for data operations, context segregation, query specification and execution, heterogeneity, main components, demonstration
Identifying, relating, consisting and querying large heterogeneous RDF sources [164]	PhD thesis Univ. Leipzig	2021	–	ADERIS, ANAPSID, **Comunica**, **CostFed**, **DARQ**, DAW, **FEDRA**, **FedX**, LDQPS, Lusail, MULDER, **Odissey**, **SaGe**, **SemaGrow**, SIHJoin, SOUIN, **SPLENDID**, TPF Client, WimuQ, WoDQA	SPARQL query federation, link traversal based SPARQL federation, dataset identification, source selection, duplicate awareness

Table 10

Academic systems selection. *C.A.* = code availability; *Sur* = mentioned by Table 9 surveys; *Het* = data heterogeneity; *Sec* = data security; *Cit.* = citations (2022/06/07); **bold** = systems selected for our survey

	System	C.A.	Relevance			# Cit.	Publications
			Sur	Het	Sec		
⩽ 2008	**DARQ**	✓	✓$_{12}$	–	–	>500	ESWC 2008 [45]
	Mediator SAIL	–	✓$_{1}$	–	–	71	IJWET 2005 [165]
	Distributed SPARQL	–	✓$_{1}$	–	–	36	ISWC 2008 (poster&demos) [166]
	Networked Graphs	–	✓$_{1}$	–	–	133	WWW 2008 [167]
	SemWIQ	–	✓$_{2}$	–	–	133	ESWC 2008 [168]
2009 - 2011	**SPLENDID**	✓	✓$_{16}$	–	–	307	COLD 2011 [46]
	FeDeRate	✓	✓$_{1}$	–	–	74	BMC Bioinf. 2009 [169]
	Atlas	–	✓$_{1}$	–	–	53	Semant. Web 2010 [170]
	QTree	–	✓$_{1}$	–	–	259	WWW 2010 [171]
	LDQPS	–	✓$_{2}$	–	–	149	ISWC 2010 [172]
	ANAPSID	✓	✓$_{11}$	–	–	237	ISWC 2011 [173]
	ADERIS	–	✓$_{1}$	–	–	43	OTM 2011 [174]
	GDS	✓	✓$_{1}$	–	–	3	*tech. rep. U. Southampton 2011 [175]*
	Min-Tree BGP	✓	✓$_{1}$	–	–	24	JIST 2011 [176]
	SIHJoin	–	✓$_{2}$	–	–	84	ESWC 2011 [177]
	SPARQL-DQP	✓	✓$_{2}$	–	–	101	ESWC 2011 [178]
2012 - 2014	**HiBISCuS**	✓	✓$_{4}$	–	–	150	ESWC 2014 [66]
	Prasser et al.	–	✓$_{1}$	–	–	150	EDBT 2012 [179]
	DISMED	✓	✓$_{1}$	–	–	6	IEEE Trans. Inf. Technol. Biomed. 2012 [180]
	WoDQA	✓	✓$_{4}$	–	–	6	LDOW 2012 [181]
	GRANATUM	✓	✓$_{4}$	–	–	113	ISWC 2014 [182], JIST 2014 [183], OEDW@EKAW 2014 [184]
	LHD	–	✓$_{5}$	–	–	54	LDOW 2013 [185]
	FedSearch	–	✓$_{1}$	–	–	35	ISWC 2013 [186]
	DAW	–	✓$_{4}$	–	–	95	ISWC 2013 [139]
	SOUIN	✓	✓$_{1}$	–	–	50	SIGMOD 2013 [187]
	Avalance	–	✓$_{1}$	–	–	100	ISWC (posters&demo) 2010 [188], J. Web Semant. 2014 [189]
2015 - 2019	**FEDRA**	✓	✓$_{2}$	–	–	40	ISWC 2015 [64]
	SemaGrow	✓	✓$_{5}$	–	–	70	SEMANTICS 2015 [93]
	oLinDa	✓	–	–	–	1	J. Inf. Data Manag. 2015 [190]
	CloudMdsQL	✓	✓$_{2}$	✓	–	159	SIGMOD 2016 [35], Distr. Parall. Datab. 2016 [59], CLOSER 2016 [191]
	BigDAWG	✓	✓$_{2}$	✓	–	>500	HPEC 2017 [57], VLDB 2015 [192], SIGMOD Record 2015 [33], etc.
	SPARQL-LD	✓	–	–	–	45	TPDL 2016 [193], ISWC (posters&demo) 2016 [194], QuWeDa@ESWC 2018 [195]
	FuhSen	✓	–	–	–	25	WWW 2016 [196]
	Odyssey	✓	✓$_{2}$	–	–	50	ISWC 2017 [79]
	SAFE	✓	✓$_{1}$	–	✓	96	J. Biom. Sem. 2017 [89], SWAT5LS 2014 [197]
	Lusail	–	✓$_{3}$	–	–	41	VLDB 2017 [198], SIGMOD 2017 [199], ICDE 2017 [200]
	Myria	✓	✓$_{2}$	–	–	217	CIDR 2017 [34], SIGMOD 2014 [201]
	LILAC	✓	–	–	–	22	J. Web Semant. 2017 [196]
	BioFed	–	✓$_{1}$	–	–	44	J. Biomedical Semant. 2017 [202]
	Comunica (TPF-Client)	✓	✓$_{1}$	–	–	393	ISWC 2018 [60], J. Web Semant. 2016 [203]
	CostFed	✓	✓$_{4}$	–	–	40	SEMANTiCS 2018 [61], ISWC (posters&demos) 2017 [204]
	BOUNCER	✓	–	–	✓	6	DEXA 2018 [205]
	SPARQL Micro-Services	✓	–	–	–	18	LDOW@WWW 2018 [206]
	PolyWeb	✓	–	✓	–	40	IEEE Access 2019 [32], Big Data 2017 [85]
	Ontario	✓	–	✓	–	65	DEXA 2019 [28], DEXA 2017 [80]
	SAGE	✓	✓$_{1}$	–	–	39	WWW 2019 [90]
	Squerall	✓	–	✓	–	58	ISWC 2019 [96], WWW 2019 [207], iiWAS 2019 [208], ISWC (satellites) 2019 [207]
	FMQO	✓	–	–	–	9	APWeb 2019 [209], DASFAA 2018 [210]
	WimuQ	✓	✓$_{1}$	–	–	6	K-CAP 2019 [211]
	SpecINT	✓	–	–	–	2	Semant. Web 2019 [212]
	LDaQ	✓	–	–	–	4	SAC 2019 [213]
	Polystore++	✓	–	✓	–	2	ICDCS 2019 [214]
	SemKT	✓	–	–	–	8	IEEE Big Data 2019 [215]
> 2020	**Obi-Wan**	✓	–	✓	–	25	EDBT 2020 [75], VLDB 2020 [76]
	Onto-Kit	✓	–	✓	–	6	FGCS 2020 [81]
	FLiQue	✓	–	–	–	1	BDA 2020 [216]

B.4. Full bibliography of the collected publications

[217] Q. Ge, P. Peng, Z. Xu, L. Zou and Z. Qin, FMQO: A Federated RDF System Supporting Multi-query Optimization, in: *Web and Big Data - Third International Joint Conference, APWeb-WAIM 2019, Chengdu, China, August 1-3, 2019, Proceedings, Part II*, J. Shao, M.L. Yiu, M. Toyoda, D. Zhang, W. Wang and B. Cui, eds, Lecture Notes in Computer Science, Vol. 11642, Springer, 2019, pp. 397–401. doi:10.1007/978-3-030-26075-0_30.

[218] I. Abdelaziz, E. Mansour, M. Ouzzani, A. Aboulnaga and P. Kalnis, Query Optimizations over Decentralized RDF Graphs, in: *33rd IEEE International Conference on Data Engineering, ICDE 2017, San Diego, CA, USA, April 19-22, 2017*, IEEE Computer Society, 2017, pp. 139–142. doi:10.1109/ICDE.2017.59.

[219] E. Mansour, I. Abdelaziz, M. Ouzzani, A. Aboulnaga and P. Kalnis, A Demonstration of Lusail: Querying Linked Data at Scale, in: *Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD Conference 2017 (Demonstrations), Chicago, IL, USA, May 14-19, 2017*, S. Salihoglu, W. Zhou, R. Chirkova, J. Yang and D. Suciu, eds, ACM, 2017, pp. 1603–1606. doi:10.1145/3035918.3058731.

[220] A.J. Elmore, J. Duggan, M. Stonebraker, M. Balazinska, U. Çetintemel, V. Gadepally, J. Heer, B. Howe, J. Kepner, T. Kraska, S. Madden, D. Maier, T.G. Mattson, S. Papadopoulos, J. Parkhurst, N. Tatbul, M. Vartak and S. Zdonik, A Demonstration of the BigDAWG Polystore System, *Proc. VLDB Endow.* **8**(12) (2015), 1908–1911. doi:10.14778/2824032.2824098. http://www.vldb.org/pvldb/vol8/p1908-Elmore.pdf.

[221] J. Zemánek and S. Schenk, Optimizing SPARQL Queries over Disparate RDF Data Sources through Distributed Semi-Joins, in: *Proceedings of the Poster and Demonstration Session at the 7th International Semantic Web Conference (ISWC), Karlsruhe, Germany, October 28, 2008*, C. Bizer and A. Joshi, eds, CEUR Workshop Proceedings, Vol. 401, CEUR-WS.org, 2008. http://ceur-ws.org/Vol-401/iswc2008pd_submission_69.pdf.

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

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

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

[225] 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.

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

[227] 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.

[228] 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.

[229] 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.

[230] 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.

[231] 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.

[232] 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.

[233] 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.

[234] 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.

[235] 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.

[236] 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.

[237] 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.

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

[239] 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).

[240] 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).
[241] 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).
[242] L. Heling and M. Acosta, A Framework for Federated SPARQL Query Processing over Heterogeneous Linked Data Fragments, *CoRR* **abs/2102.03269** (2021).
[243] G. Montoya, H. Skaf-Molli, P. Molli and M. Vidal, Fedra: Query Processing for SPARQL Federations with Divergence, *CoRR* **abs/1407.2899** (2014).
[244] 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).
[245] 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.
[246] 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.
[247] X. Wang, L.M. Haas and A. Meliou, Explaining Data Integration, *IEEE Data Eng. Bull.* **41**(2) (2018), 47–58.
[248] 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.
[249] M.T. Özsu, A survey of RDF data management systems, *Frontiers Comput. Sci.* **10**(3) (2016), 418–432. doi:10.1007/s11704-016-5554-y.
[250] 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.
[251] 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.
[252] M.V. Sande, R. Verborgh, A. Dimou, P. Colpaert and E. Mannens, Hypermedia-Based Discovery for Source Selection Using Low-Cost Linked Data Interfaces, *Int. J. Semantic Web Inf. Syst.* **12**(3) (2016), 79–110. doi:10.4018/IJSWIS.2016070103.
[253] 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.
[254] 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.
[255] 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.
[256] 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.
[257] 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.
[258] 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.
[259] 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.
[260] 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.
[261] 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.
[262] 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.
[263] 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.
[264] 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.
[265] 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.
[266] 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.
[267] 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.
[268] 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.
[269] 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.

[270] 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.
[271] 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.
[272] A. Stolpe, A logical characterisation of SPARQL federation, *Semantic Web* **6**(6) (2015), 565–584. doi:10.3233/SW-140160.
[273] N.F. Noy, Semantic Integration: A Survey Of Ontology-Based Approaches, *SIGMOD Rec.* **33**(4) (2004), 65–70. doi:10.1145/1041410.1041421.
[274] 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.
[275] 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.
[276] 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.
[277] 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.
[278] 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.
[279] 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.
[280] 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.
[281] 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.
[282] 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.
[283] 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.
[284] 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.
[285] Z. Kaoudi, M. Koubarakis, K. Kyzirakos, I. Miliaraki, M. Magiridou and A. Papadakis-Pesaresi, Atlas: Storing, updating and querying RDF(S) data on top of DHTs, *J. Web Semant.* **8**(4) (2010), 271–277. doi:10.1016/j.websem.2010.07.001.
[286] M. Acosta, E. Simperl, F. Flöck and M.-E. Vidal, Enhancing answer completeness of SPARQL queries via crowdsourcing, *J. Web Semantics* **45** (2017), 41–62. doi:10.1016/j.websem.2017.07.001.
[287] R. Verborgh, M.V. Sande, O. Hartig, J.V. Herwegen, L.D. Vocht, B.D. Meester, G. Haesendonck and P. Colpaert, Triple Pattern Fragments: A low-cost knowledge graph interface for the Web, *J. Web Semant.* **37-38** (2016), 184–206. doi:10.1016/j.websem.2016.03.003.
[288] 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.
[289] 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.
[290] O. Golovnin, Data federation through on-demand queries in intelligent transport systems **1694**(1) (2020), 012030, IOP Publishing.
[291] L. Heling and M. Acosta, A Framework for Federated SPARQL Query Processing over Heterogeneous Linked Data Fragments, *arXiv preprint arXiv:2102.03269* (2021).
[292] 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.
[293] 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).
[294] Y. Khan and R. Sahay, SPARQL Query Federation over Biomedical Data, *Insight Centre for Data Analytics, National University Literature* (2019).
[295] 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.
[296] 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.
[297] 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.
[298] 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.
[299] M. Mountantonakis and Y. Tzitzikas, Large-scale semantic integration of linked data: A survey, *ACM Computing Surveys (CSUR)* **52**(5) (2019), 1–40.
[300] 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.

[301] N.A. Rakhmawati, An Holistic Evaluation of Federated SPARQL Query Engine, in: *Proc. of Information Systems International Conference (ISICO)*, 2013.
[302] N.A. Rakhmawati et al., How interlinks influence federated over sparql endpoints, *International Journal of Internet and Distributed Systems* **1**(01) (2013), 1.
[303] N.A. Rakhmawati and L.N. Fadzilah, Dataset characteristics identification for federated sparql query, *Scientific Journal of Informatics* **6**(1) (2019), 23–33.
[304] 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).
[305] 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.
[306] 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).
[307] 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.
[308] P. Szabo, Data Virtualization and Federation, *Stone Bond Technologies* (2014).
[309] R. Van Der Lans, *Data Virtualization for business intelligence systems: revolutionizing data integration for data warehouses*, Elsevier, 2012.
[310] 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.
[311] 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.
[312] 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.
[313] 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.
[314] 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.
[315] 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.
[316] 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.
[317] O. Mora, G. Engelbrecht and J. Bisbal, A Service-Oriented Distributed Semantic Mediator: Integrating Multiscale Biomedical Information, *IEEE Trans. Inf. Technol. Biomed.* **16**(6) (2012), 1296–1303. doi:10.1109/TITB.2012.2215045.
[318] H. Stuckenschmidt, R. Vdovjak, J. Broekstra and G. Houben, Towards distributed processing of RDF path queries, *Int. J. Web Eng. Technol.* **2**(2/3) (2005), 207–230. doi:10.1504/IJWET.2005.008484.
[319] J. Ahn, J. Eom, S. Nam, N. Zong, D. Im and H. Kim, xStore: Federated temporal query processing for large scale RDF triples on a cloud environment, *Neurocomputing* **256** (2017), 5–12. doi:10.1016/j.neucom.2016.03.116.
[320] L. Martín, A. Anguita, V. Maojo, E. Bonsma, A.I.D. Bucur, J. Vrijnsen, M. Brochhausen, C. Cocos, H. Stenzhorn, M. Tsiknakis, M. Doerr and H. Kondylakis, Ontology Based Integration of Distributed and Heterogeneous Data Sources in ACGT, in: *Proceedings of the First International Conference on Health Informatics, HEALTHINF 2008, Funchal, Madeira, Portugal, January 28-31, 2008, Volume 1*, L. Azevedo and A.R. Londral, eds, INSTICC - Institute for Systems and Technologies of Information, Control and Communication, 2008, pp. 301–306.
[321] 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.
[322] 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.
[323] A. Bogdanov, A. Degtyarev, N. Shchegoleva, V. Korkhov and V. Khvatov, Big Data Virtualization: Why and How?, in: *Proceedings of the 4th International Workshop on Data Life Cycle in Physics (DLC)*, CEUR Workshop Proceedings (2679), 2020, pp. 11–21.
[324] 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.
[325] 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.
[326] 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.
[327] 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.

[328] P.N. Sawadogo, É. Scholly, C. Favre, É. Ferey, S. Loudcher and J. Darmont, Metadata Systems for Data Lakes: Models and Features, in: *Proceedings of 1st International Workshop on BI and Big Data Applications (BBIGAP) co-located with ADBIS 2019, 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.

[329] 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.

[330] 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.

[331] 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.

[332] 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.

[333] 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.

[334] 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.

[335] 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.

[336] 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.

[337] 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.

[338] 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.

[339] 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.

[340] 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.

[341] 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.

[342] 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.

[343] 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.

[344] 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.

[345] 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.

[346] 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.

[347] 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.

[348] 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.

[349] P.D. Rohde and M. Vidal, Optimizing Federated Queries Based on the Physical Design of a Data Lake, in: *Proceedings of the Workshop on Search, Exploration, and Analysis in Heterogeneous Datastores (SEA Data), colocated with EDBT/ICDT 2020 Joint Conference, Copenhagen, Denmark, March 30, 2020*, CEUR Workshop Proceedings, Vol. 2578, CEUR-WS.org, 2020.

[350] 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.

[351] 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.

[352] 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.

[353] 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.

[354] 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.

[355] 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.

[356] 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.

[357] 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.

[358] 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.

[359] 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.

[360] 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.

[361] 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.

[362] 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.

[363] 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.

[364] 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.

[365] 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.

[366] T. Minier, G. Montoya, H. Skaf-Molli and P. Molli, PeNeLoop: Parallelizing Federated SPARQL Queries in Presence of Replicated Fragments, in: *Proceedings of 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.

[367] 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.

[368] 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.

[369] 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.

[370] 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.

[371] 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.

[372] 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.

[373] 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.

[374] 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.

[375] 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 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.

[376] 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.

[377] 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.

[378] 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.

[379] 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.

[380] 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.

[381] 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.

[382] S. Jaiswal and M. Lefrançois, Towards Federated Queries for Web of Things Devices, in: *Proceedings of Workshop on Semantic Interoperability and Standardization in the IoT (SIS-IoT) 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.

[383] 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.

[384] 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.

[385] 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.

[386] K. Schlegel, F. Stegmaier, S. Bayerl, M. Granitzer and H. Kosch, Balloon Fusion: SPARQL rewriting based on unified co-reference information, in: *Proceedings of 5th International Workshop on Data Engineering meets the Semantic Web (DESWeb) co-located with the 30th International Conference on Data Engineering, ICDE 2014, Chicago, IL, USA, March 31 - April 4, 2014*, IEEE Computer Society, 2014, pp. 254–259. doi:10.1109/ICDEW.2014.6818335.

[387] 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.

[388] 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.

[389] 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.

[390] 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.

[391] 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.

[392] 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.

[393] 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.

[394] 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.

[395] 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.

[396] 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.

[397] 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.

[398] 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.

[399] 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.

[400] 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.

[401] 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.

[402] 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.

[403] 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.

[404] 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.

[405] G. Ladwig and T. Tran, Linked Data Query Processing Strategies, in: *The Semantic Web - ISWC 2010 - 9th International Semantic Web Conference, ISWC 2010, Shanghai, China, November 7-11, 2010, Revised Selected Papers, Part I*, P.F. Patel-Schneider, Y. Pan, P. Hitzler, P. Mika, L. Zhang, J.Z. Pan, I. Horrocks and B. Glimm, eds, Lecture Notes in Computer Science, Vol. 6496, Springer, 2010, pp. 453–469. doi:10.1007/978-3-642-17746-0_29.

[406] 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.

[407] 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.

[408] 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.

[409] 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.

[410] 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.

[411] 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.

[412] 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.

[413] 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.

[414] 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.

[415] 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.

[416] 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.

[417] 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.

[418] 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.

[419] 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.

[420] 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.

[421] 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.

[422] 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.

[423] 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.

[424] 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.

[425] 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.

[426] 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.

[427] 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.

[428] 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.

[429] 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.

[430] 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.

[431] 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.

[432] 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.

[433] 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.

[434] 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.

[435] 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.

[436] 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.

[437] 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.

[438] 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.

[439] 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 (Demonstrations), Snowbird, UT, USA, June 22-27, 2014*, ACM, 2014, pp. 881–884. doi:10.1145/2588555.2594530.

[440] 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.

[441] 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.

[442] 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.

[443] 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.

[444] 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.

[445] 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.

[446] 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.

[447] 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.

[448] 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.

[449] 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.

[450] 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.

[451] 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.

[452] 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.

[453] 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.

[454] 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.

[455] 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.

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

[457] 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.

[458] 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.

[459] 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.

[460] 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.

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

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

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

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

[465] 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.

[466] 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.
[467] 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.
[468] 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.
[469] 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.
[470] 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.
[471] 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.
[472] P. Molli, H. Skaf-Molli and A. Grall, SemCat: Source Selection Services for Linked Data, PhD thesis, université de Nantes, 2020.
[473] J. Pålsson, Querying Federations of Eiffel Event Data Repositories, 2020.
[474] 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.
[475] P.D. Rohde, Query Optimization Techniques For Scaling Up To Data Variety, Master's thesis, Hannover: Institutionelles Repositorium der Leibniz Universität Hannover, 2019.
[476] C.R. da Silva Teixeira, Implementation of a data virtualization laver applied to insurance data, Master's thesis, University of Porto, 2016.
[477] M. Wigham, State of the art in federated querying in SPARQL, Technical Report, Wageningen UR, 2014.
[478] L. Xu, New capabilities for large-scale exploratory data analysis, PhD thesis, University of Illinois at Urbana-Champaign, 2020.
[479] P. Serrano-Alvarado, Protecting user data in distributed systems, PhD thesis, Université de Nantes (UN), 2020.
[480] A. Harth, J. Umbrich, A. Hogan and S. Decker, YARS2: A Federated Repository for Querying Graph Structured Data from the Web, in: *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*, K. Aberer, K. Choi, N.F. Noy, D. Allemang, K. Lee, L.J.B. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber and P. Cudré-Mauroux, eds, Lecture Notes in Computer Science, Vol. 4825, Springer, 2007, pp. 211–224. doi:10.1007/978-3-540-76298-0_16.
[481] S. Lynden, I. Kojima, A. Matono and Y. Tanimura, Aderis: An adaptive query processor for joining federated sparql endpoints, in: *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, 2011, pp. 808–817.
[482] G. Ladwig and T. Tran, SIHJoin: Querying Remote and Local Linked Data, in: *The Semantic Web: Research and Applications - 8th Extended Semantic Web Conference, ESWC 2011, Heraklion, Crete, Greece, May 29-June 2, 2011, Proceedings, Part I*, G. Antoniou, M. Grobelnik, E.P.B. Simperl, B. Parsia, D. Plexousakis, P.D. Leenheer and J.Z. Pan, eds, Lecture Notes in Computer Science, Vol. 6643, Springer, 2011, pp. 139–153. doi:10.1007/978-3-642-21034-1_10.
[483] X. Wang, T. Tiropanis and H.C. Davis, LHD: Optimising Linked Data Query Processing Using Parallelisation, in: *Proceedings of the WWW2013 Workshop on Linked Data on the Web, Rio de Janeiro, Brazil, 14 May, 2013*, C. Bizer, T. Heath, T. Berners-Lee, M. Hausenblas and S. Auer, eds, CEUR Workshop Proceedings, Vol. 996, CEUR-WS.org, 2013. http://ceur-ws.org/Vol-996/papers/ldow2013-paper-06.pdf.
[484] O. Hartig, SQUIN: a traversal based query execution system for the web of linked data, in: *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2013, New York, NY, USA, June 22-27, 2013*, K.A. Ross, D. Srivastava and D. Papadias, eds, ACM, 2013, pp. 1081–1084. doi:10.1145/2463676.2465231.
[485] O. Görlitz, Distributed query processing for federated RDF data management, PhD thesis, University of Koblenz-Landau, 2015. http://kola.opus.hbz-nrw.de/volltexte/2015/1091/pdf/diss_print_final.pdf.
[486] S. Konstantopoulos, A. Charalambidis, A. Troumpoukis, G. Mouchakis and V. Karkaletsis, The Sevod Vocabulary for Dataset Descriptions for Federated Querying, 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*, E. Demidova, S. Dietze, J. Szymanski and J.G. Breslin, eds, CEUR Workshop Proceedings, Vol. 1927, CEUR-WS.org, 2017. http://ceur-ws.org/Vol-1927/paper4.pdf.
[487] G.D. Giacomo, D. Lembo, M. Lenzerini, A. Poggi and R. Rosati, Using Ontologies for Semantic Data Integration, in: *A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years*, S. Flesca, S. Greco, E. Masciari and D. Saccà, eds, Studies in Big Data, Vol. 31, Springer International Publishing, 2018, pp. 187–202. doi:10.1007/978-3-319-61893-7_11.
[488] T. Minier, H. Skaf-Molli and P. Molli, SaGe: Web Preemption for Public SPARQL Query Services, in: *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, L. Liu, R.W. White, A. Mantrach, F. Silvestri, J.J. McAuley, R. Baeza-Yates and L. Zia, eds, ACM, 2019, pp. 1268–1278. doi:10.1145/3308558.3313652.
[489] K.S. Aggour, V.S. Kumar, P. Cuddihy, J.W. Williams, V. Gupta, L. Dial, T. Hanlon, J. Gambone and J. Vinciquerra, Federated Multimodal Big Data Storage & Analytics Platform for Additive Manufacturing, in: *2019 IEEE International Conference on Big Data (IEEE BigData), Los Angeles, CA, USA, December 9-12, 2019*, C.K. Baru, J. Huan, L. Khan, X. Hu, R. Ak, Y. Tian, R.S. Barga, C. Zaniolo, K. Lee and Y.F. Ye, eds, IEEE, 2019, pp. 1729–1738. doi:10.1109/BigData47090.2019.9006495.
[490] M. Belcao, E. Falzone, E. Bionda and E.D. Valle, Chimera: A Bridge Between Big Data Analytics and Semantic Technologies, in: *The Semantic Web - ISWC 2021 - 20th International Semantic Web Conference, ISWC 2021, Virtual Event, October 24-28, 2021, Proceedings*, A. Hotho, E. Blomqvist, S. Dietze, A. Fokoue, Y. Ding, P.M. Barnaghi, A. Haller, M. Dragoni and H. Alani, eds, Lecture Notes in Computer Science, Vol. 12922, Springer, 2021, pp. 463–479. doi:10.1007/978-3-030-88361-4_27.

[491] K.M. Endris, M. Vidal and D. Graux, Federated Query Processing, in: *Knowledge Graphs and Big Data Processing*, V. Janev, D. Graux, H. Jabeen and E. Sallinger, eds, Lecture Notes in Computer Science, Vol. 12072, Springer, 2020, pp. 73–86. doi:10.1007/978-3-030-53199-7_5.

[492] L. Heling and M. Acosta, Cost- and Robustness-Based Query Optimization for Linked Data Fragments, in: *The Semantic Web - ISWC 2020 - 19th International Semantic Web Conference, Athens, Greece, November 2-6, 2020, Proceedings, Part I*, J.Z. Pan, V.A.M. Tamma, C. d'Amato, K. Janowicz, B. Fu, A. Polleres, O. Seneviratne and L. Kagal, eds, Lecture Notes in Computer Science, Vol. 12506, Springer, 2020, pp. 238–257. doi:10.1007/978-3-030-62419-4_14.

[493] M. Saleem, Y. Khan, A. Hasnain, I. Ermilov and A.-C. Ngonga Ngomo, An evaluation of SPARQL federation engines over multiple endpoints, in: *International Web Conference*, 2018.

[494] S. Schenk and S. Staab, Networked graphs: a declarative mechanism for SPARQL rules, SPARQL views and RDF data integration on the web, in: *Proceedings of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21-25, 2008*, J. Huai, R. Chen, H. Hon, Y. Liu, W. Ma, A. Tomkins and X. Zhang, eds, ACM, 2008, pp. 585–594. doi:10.1145/1367497.1367577.

[495] X. Wang, T. Tiropanis and H.C. Davis, Evaluating Graph Traversal Algorithms for Distributed SPARQL Query Optimization, in: *The Semantic Web - Joint International Semantic Technology Conference, JIST 2011, Hangzhou, China, December 4-7, 2011. Proceedings*, J.Z. Pan, H. Chen, H. Kim, J. Li, Z. Wu, I. Horrocks, R. Mizoguchi and Z. Wu, eds, Lecture Notes in Computer Science, Vol. 7185, Springer, 2011, pp. 210–225. doi:10.1007/978-3-642-29923-0_14.

[496] F. Prasser, A. Kemper and K.A. Kuhn, Efficient distributed query processing for autonomous RDF databases, in: *15th International Conference on Extending Database Technology, EDBT '12, Berlin, Germany, March 27-30, 2012, Proceedings*, E.A. Rundensteiner, V. Markl, I. Manolescu, S. Amer-Yahia, F. Naumann and I. Ari, eds, ACM, 2012, pp. 372–383. doi:10.1145/2247596.2247640.

[497] Q. Mehmood, A. Jha, D. Rebholz-Schuhmann and R. Sahay, FedS: Towards Traversing Federated RDF Graphs, in: *Big Data Analytics and Knowledge Discovery - 20th International Conference, DaWaK 2018, Regensburg, Germany, September 3-6, 2018, Proceedings*, C. Ordonez and L. Bellatreche, eds, Lecture Notes in Computer Science, Vol. 11031, Springer, 2018, pp. 34–45. doi:10.1007/978-3-319-98539-8_3.

[498] N. Ge, Z. Qin, P. Peng and L. Zou, FedTopK: Top-K Queries Optimization over Federated RDF Systems, in: *Database Systems for Advanced Applications - 26th International Conference, DASFAA 2021, Taipei, Taiwan, April 11-14, 2021, Proceedings, Part III*, C.S. Jensen, E. Lim, D. Yang, W. Lee, V.S. Tseng, V. Kalogeraki, J. Huang and C. Shen, eds, Lecture Notes in Computer Science, Vol. 12683, Springer, 2021, pp. 595–599. doi:10.1007/978-3-030-73200-4_42.

[499] G. Rao, B. Zhao, X. Zhang and Z. Feng, FedQL: A Framework for Federated Queries Processing on RDF Stream and Relational Data, in: *Database Systems for Advanced Applications - DASFAA 2018 International Workshops: BDMS, BDQM, GDMA, and SeCoP, Gold Coast, QLD, Australia, May 21-24, 2018, Proceedings*, C. Liu, L. Zou and J. Li, eds, Lecture Notes in Computer Science, Vol. 10829, Springer, 2018, pp. 141–155. doi:10.1007/978-3-319-91455-8_14.

[500] K.M. Endris, Z. Almhithawi, I. Lytra, M. Vidal and S. Auer, BOUNCER: Privacy-Aware Query Processing over Federations of RDF Datasets, in: *Database and Expert Systems Applications - 29th International Conference, DEXA 2018, Regensburg, Germany, September 3-6, 2018, Proceedings, Part I*, S. Hartmann, H. Ma, A. Hameurlain, G. Pernul and R.R. Wagner, eds, Lecture Notes in Computer Science, Vol. 11029, Springer, 2018, pp. 69–84. doi:10.1007/978-3-319-98809-2_5.

[501] D. Collarana, C. Lange and S. Auer, FuhSen: A Platform for Federated, RDF-based Hybrid Search, in: *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11-15, 2016, Companion Volume*, J. Bourdeau, J. Hendler, R. Nkambou, I. Horrocks and B.Y. Zhao, eds, ACM, 2016, pp. 171–174. doi:10.1145/2872518.2890535.

[502] A. Fatima, C. Luca, G. Wilson and M.S. Kettouch, Result Optimisation for Federated SPARQL Queries, in: *UKSim-AMSS 17th International Conference on Computer Modelling and Simulation, UKSim 2015, Cambridge, United Kingdom, March 25-27, 2015*, D. Al-Dabass, A. Orsoni, R.J. Cant, Z. Ibrahim and I. Saad, eds, IEEE, 2015, pp. 491–496. doi:10.1109/UKSim.2015.50.

[503] K. Makris, N. Gioldasis, N. Bikakis and S. Christodoulakis, Ontology Mapping and SPARQL Rewriting for Querying Federated RDF Data Sources - (Short Paper), in: *On the Move to Meaningful Internet Systems, OTM 2010 - Confederated International Conferences: CoopIS, IS, DOA and ODBASE, Hersonissos, Crete, Greece, October 25-29, 2010, Proceedings, Part II*, R. Meersman, T.S. Dillon and P. Herrero, eds, Lecture Notes in Computer Science, Vol. 6427, Springer, 2010, pp. 1108–1117. doi:10.1007/978-3-642-16949-6_32.

[504] G. Montoya, M. Vidal and M. Acosta, DEFENDER: A DEcomposer for quEries agaiNst feDERations of Endpoints, in: *The Semantic Web: ESWC 2012 Satellite Events - ESWC 2012 Satellite Events, Heraklion, Crete, Greece, May 27-31, 2012. Revised Selected Papers*, E. Simperl, B. Norton, D. Mladenic, E.D. Valle, I. Fundulaki, A. Passant and R. Troncy, eds, Lecture Notes in Computer Science, Vol. 7540, Springer, 2012, pp. 480–484. doi:10.1007/978-3-662-46641-4_49.

[505] Q. Ge, P. Peng, Z. Xu, L. Zou and Z. Qin, FMQO: A Federated RDF System Supporting Multi-query Optimization, in: *Proc. of Int. Joint Conf. APWeb-WAIM*, LNCS, Vol. 11642, Springer, 2019, pp. 397–401. doi:10.1007/978-3-030-26075-0_30.

[506] 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.

[507] A. Hasnain, R. Fox, S. Decker and H.F. Deus, Cataloguing and Linking Life Sciences LOD Cloud, in: *Proc. of 1st Int. Workshop on Ontology Engineering in a Data-driven World (OEDW), co-located with EKAW*, 2012.

[508] A. Hasnain, S.S. e Zainab, M.R. Kamdar, Q. Mehmood, C.N.W. Jr., Q.A. Fatimah, H.F. Deus, M. Mehdi and S. Decker, A Roadmap for Navigating the Life Sciences Linked Open Data Cloud, in: *Semantic Technology - 4th Joint International Conference, JIST 2014, Chiang Mai, Thailand, November 9-11, 2014. Revised Selected Papers*, T. Supnithi, T. Yamaguchi, J.Z. Pan, V. Wuwongse and M. Buranarach, eds, Lecture Notes in Computer Science, Vol. 8943, Springer, 2014, pp. 97–112. doi:10.1007/978-3-319-15615-6_8.

[509] A. Hasnain, M.R. Kamdar, P. Hasapis, D. Zeginis, C.N.W. Jr., H.F. Deus, D. Ntalaperas, K.A. Tarabanis, M. Mehdi and S. Decker, Linked Biomedical Dataspace: Lessons Learned Integrating Data for Drug Discovery, in: *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I*, P. Mika, T. Tudorache, A. Bernstein, C. Welty, C.A. Knoblock, D. Vrandecic, P. Groth, N.F. Noy, K. Janowicz and C.A. Goble, eds, Lecture Notes in Computer Science, Vol. 8796, Springer, 2014, pp. 114–130. doi:10.1007/978-3-319-11964-9_8.

[510] A. Harth, K. Hose, M. Karnstedt, A. Polleres, K. Sattler and J. Umbrich, Data summaries for on-demand queries over linked data, in: *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, M. Rappa, P. Jones, J. Freire and S. Chakrabarti, eds, ACM, 2010, pp. 411–420. doi:10.1145/1772690.1772733.

[511] X. Wang, T. Tiropanis and H.C. Davis, Querying the Web of Data with Graph Theory-based Techniques, Technical Report, University of Southampton, 2011.