

# Balancing RDF generation from heterogeneous data sources

Dylan Van Assche \*

IDLab, Department of Electronics and Information Systems,  
Ghent University – imec, Technologiepark-Zwijnaarde 122, 9052 Ghent, Belgium  
`dylan.vanassche@ugent.be`

**Abstract.** Knowledge graphs in RDF are often generated from heterogeneous data sources to power services. However, knowledge graph generation is an unbalanced effort for producers compared to consumers of a knowledge graph. In this paper, I present my research about (i) investigating current RDF knowledge graph production and consumption approaches, and (ii) how to involve the consumer into a hybrid RDF generation approach to reduce the necessary resources for generating RDF for producers & consumers. I discuss the shortcomings of existing approaches for RDF generation from heterogeneous data sources (i.e., materialization and virtualization) and how I will address these: a Systematic Literature Review; an analysis and a set of guidelines for producers to select the right approach for an use case; and a combined hybrid approach to balance the producer’s and consumer’s effort in RDF generation. I already performed a Systematic Literature Review to get an overview of the existing approaches for RDF production from heterogeneous data sources. These results will be used to establish a set of producer guidelines, a benchmark to compare the current materialization and virtualization approaches, and evaluate the proposed hybrid approach. Thanks to my research, knowledge graph production and consumption will be more balanced and accessible to smaller companies and individuals. This way, they can focus on providing better services on top of a knowledge graph instead of being limited by the lack of computing resources to harvest enormous amounts of data from the Web and integrate it into a knowledge graph.

## 1 Introduction

Over the past two decades, several RDF generation approaches emerged such as materialization & virtualization. On the one hand, materialization approaches in the form of Extract-Transform-Load (ETL) [1] extract data from heterogeneous data sources, transform and integrate them completely or partially into a knowledge graph, and materialize it to a certain target, such as a triple store, a file, etc. On the other hand, virtualization in the form of Ontology Based Data Access (OBDA) [2] provides query access to a virtual knowledge graph on top of

---

\* Supervised by Anastasia Dimou  & Ben De Meester .

the heterogeneous data. OBDA allows consumers to ask SPARQL queries and get an tailored answer for their query. Only the data necessary to answer the query is used and transformed by the producer.

Currently, producers are solely responsible for generating knowledge graphs. There is no way for a producer to determine which RDF generation approach is the most suitable depending on e.g., the properties of the data or how the RDF is used by consumers. Moreover, integrating enormous amounts of data into RDF is unfeasible since the producer’s and consumer’s effort is unbalanced.

No guidelines exist for producers to select the right approach depending on how their generated RDF is consumed. Therefore, producers cannot optimize their RDF generation, even though approaches exist which optimize parts of it.

RDF producers and consumers aim to minimize their own effort for producing and consuming a knowledge graph, but these efforts are unbalanced in favour of consumers. Producers are responsible for generating RDF from heterogeneous data sources and answering queries from consumers. Consumers ask queries and consume the answers as RDF, provided by the producer. Consequently, producers not only provide the data but also need to provide the most resources compared to consumers. Alternative approaches involving the consumer in the generation process are not investigated yet. This could better balance the efforts: consumers can contribute to the generation together with producers. This way, the producer’s effort is reduced and balanced with its consumers.

In this PhD thesis I aim to (i) investigate which key factors influence RDF generation by analyzing existing approaches, (ii) provide a set of guidelines for producers to select the right RDF generation approach for a given use case, and (iii) introduce a new approach which involves consumers during the generation process. This way, producers’ and consumers’ efforts can be balanced better. Thanks to my research, smaller companies and individuals will be able to produce and consume knowledge graphs in RDF without having to invest in a large infrastructure to harvest and integrate all data into RDF. Smaller companies and individuals can focus on building products such as virtual assistants or smart route planners instead of entering first a data harvesting competition.

## 2 State of the Art

I provide an overview of current knowledge graph generation approaches, their strengths and weaknesses, and discuss existing benchmarks for their generation.

*Knowledge graph generation* Several approaches exist to generate knowledge graphs based on materialization or virtualization, each with their own merits.

ETL approaches transform all data from heterogeneous data sources into a materialized knowledge graph. Several approaches exist based on the R2RML mapping language [3], e.g. KR2RML [4] or Morph-xR2RML [5], or its extensions, e.g., RMLMapper [6]. Some RML-based ETL approaches optimize the mapping rules execution, e.g., SDM-RDFizer [7] by avoiding duplicates, Morph-

KGC<sup>1</sup> through mapping rules partitioning, or Morph-CSV [8] by normalizing and cleaning tabular data. Streaming-based ETL approaches, e.g., RML-Streamer [9] process large heterogeneous data sources in a streaming way. Besides R2RML based, SPARQL query language [10] based approaches exists as well, e.g., SPARQL-Generate [11], SPARQL-Anything [12], or XSPARQL [13]. They repurposed SPARQL to generate RDF from heterogeneous data sources; so does ShExML [14] which repurposes the constraint language ShEx [15].

OBDA approaches, e.g., Morph [16], UltraWrap [17], or Virtuoso<sup>2</sup> answer consumers' SPARQL queries over a virtual knowledge graph from a homogeneous data source, e.g., relational database. The response is generated at query time from a single data source. Recently, OBDA approaches such as Ontop [18], Squerall [19], Ontario [20], or PolyWeb [21] emerged for heterogeneous data.

Both ETL and OBDA approaches answer consumers' queries, but differ in their execution. OBDA provides a query interface for consumers, whereas ETL relies on external RDF triple stores for query executing. If a knowledge graph generated by ETL must be changed, the whole knowledge graph is regenerated. Depending on the data sources, this may take significant resources and execution time. If the data sources change faster than the knowledge graph is regenerated, these changes may not even appear in the generated RDF depending on the frequency of the regeneration process. However, this is not the case for OBDA, as the RDF is generated for each query from the data sources. This way, the generated RDF always has the data changes incorporated. Execution time for both approaches may heavily increase depending on e.g., the query and the size of the data sources. Scalability depends on the RDF generation approach, query execution, query type, size of the data sources, and how frequently they change.

*Benchmarks* Over the past decade, several benchmarks were proposed to evaluate and compare knowledge graph generation approaches. Benchmarks such as GTFS-Madrid-Bench [22], Berlin SPARQL Benchmark (BSBM) [23], Lehigh University Benchmark (LUBM) [24], SP<sup>2</sup>Bench [25], LSLOD [26], DBpedia SPARQL Benchmark [27], Linked Open Data Integration Benchmark (LODIB) [28], or Norwegian Petroleum Directorate Benchmark (NPD) [29] focus on evaluating virtualization approaches but no materialization approaches, as they provide a set of SPARQL queries to be executed by the virtualization query engine.

### 3 Problem Statement and Contributions

Both ETL materialization and OBDA virtualization for RDF generation from heterogeneous data sources are computationally intensive operations depending on factors e.g. available computing resources, data freshness, etc. These factors and approaches combining both are not investigated yet. **Research Question:** *How can RDF be generated in balanced way for producers and consumers with respect to execution time, computing resources, and consumers' queries?*

<sup>1</sup> <https://github.com/oeg-upm/morph-kgc>

<sup>2</sup> <https://virtuoso.openlinksw.com/>

**Hypothesis** When producers and consumers collaborate during the knowledge graph generation, a knowledge graph will be generated faster, with less computing resources e.g. CPU, RAM, storage, and network bandwidth, and tailored towards answering queries from consumers.

I split my Research Question (RQ) into three subquestions: RQ1 investigates the factors influencing RDF generation from heterogeneous data sources, the State of the Art, and open issues. These factors are used as a basis for RQ2 to investigate how these factors influence existing RDF generation approaches with a benchmark. The benchmark results can be used as a base to define a set of producer guidelines to better select the right approach. In RQ3, I use the results of RQ2 to introduce a new RDF generation approach.

**Research Question 1 (RQ1):** *What are the open issues, key factors regarding computing resources or consumers' usage, and available approaches in deciding if (part of) RDF is produced through materialization or virtualization.*

**Hypothesis 1 (H1):** Several approaches exist for materialization and virtualization based on existing specifications e.g. R2RML [3], or SPARQL query language [10]. Available computing resources, data size, query type, execution time, and update frequency influence when and how an RDF graph should be produced. Several open issues remain regarding transforming heterogeneous data.

**Contribution 1:** Systematic Literature Review to determine these factors based on the last 20 years of research in this domain.

**Materialization Vs virtualization** Selecting between virtualization or materialization is highly subjective because there are currently no studies evaluating which approach is the most suitable depending on how consumers access and use the generated RDF graph. Producers are responsible for the complete RDF generation while consumers only use the generated RDF or wait for an answer from the producer for their query. Since no guidelines exist, producers cannot optimize their RDF generation depending on their own resources and RDF use.

Generating materialized and virtualized RDF is constrained with respect to execution time [30], computing resources [18, 30], bandwidth [31], performance [32], and query execution [31] because producers do not know which generation approach is the most suitable given its own resources and the RDF use. Since the producer needs to provide most resources for generating RDF from heterogeneous data sources and answering consumers' queries, guidelines for selecting the right approach are needed to minimize its effort. For instance, depending on the size of the data, producers may benefit from materialization because at the crossing point, the query and virtualized access of OBDA may cause more overhead than materializing (part of) a knowledge graph. However, this may influence how frequently a knowledge graph is updated which may affect consumers depending on how they use e.g. route changes with a route planning use case needs frequent updates while a weather prediction for next week may not. These guidelines try to provide for each key factor the trade-off when selecting a certain approach.

**Research Question 2 (RQ2):** *How influence the identified key factors the producer's effort when selecting either materialization or virtualization?*

**Hypothesis 2 (H2):** At least one crossing point exists between materialization and virtualization for each key factor. This crossing point determines a set of guidelines for producers to select the most suitable generation approach.

**Contribution 2:** Set of guidelines to select materialization or virtualization for generating RDF by broadening the GTFS-Madrid-Bench benchmark’s scope.

**Producer Vs Consumer** The producer’s effort is unbalanced compared to the consumer’s effort since the producer needs to generate RDF but also answer the consumer’s query. Moreover, production and consumption are still considered independent tasks. Each party executing one of these tasks, aims to reduce its own effort. This causes an imbalance of the producers’ and consumers’ efforts. Approaches where consumers and producers both participate in the materialization and/or virtualization process are not investigated yet.

**Research Question 3 (RQ3):** *How can consumers reduce the producer’s effort regarding execution time and computing resources when generating RDF?*

**Hypothesis 3 (H3):** The execution time and computing resources are significantly reduced for producers when consumers are involved since consumers also generate parts of the RDF instead of only the producer.

**Contribution 3:** Involving the consumer in the existing materialization and virtualization approaches for generating RDF from heterogeneous data sources to balance the effort better between producers and consumers.

## 4 Research Methodology and Approach

I execute this research in three parts, each related to a RQ, to investigate the current State of the Art in depth and find a balance between the different key factors. The Systematic Literature Review (Part 1) determined the key factors influencing knowledge graph generation such as computing resources, execution time, etc. These key factors are used in a benchmark to evaluate materialization and virtualization approaches. Based on these results and how their generated RDF is used, I introduce a set of producer guidelines (Part 2) to select the right approach for their own resources. These guidelines can be used on the results of RDF generation benchmarks such as GTFS-Madrid-Bench [22]. I use these guidelines and benchmark results to introduce a new approach (Part 3) which involves the consumer to balance the efforts between producers and consumers.

*Contribution 1: Systematic Literature Review* I systematically reviewed the literature of the last 20 years of research in this domain to establish a good overview of which approaches exist, their strengths and weaknesses, etc. This article is at the time of writing under major revision at the Journal of Web Semantics<sup>3</sup>. Relying on these results, I determined a set of key factors, e.g., data size, type of queries, or data freshness that influence an RDF graph’s generation.

*Contribution 2: Selection guidelines for materialization and virtualization* I will benchmark materialization and virtualization approaches based on the identified

<sup>3</sup> <https://www.websemanticsjournal.org/>

key factors. The results of this benchmark will be used to create a set of guidelines to select the right approach depending on available computing resources and how consumers use a generated knowledge graph. I expect that the results will show at least one crossing point between materialization and virtualization in our benchmark which allows me to define a guideline for each evaluated key factor. These guidelines can be used by producers to select the right approach based on results of RDF generation benchmarks such as GTFS-Madrid-Bench [22]. Materialization is commonly used for generating RDF from one or multiple large heterogeneous data sources. Once the RDF is materialized, it can be used to answer queries from multiple consumers without regenerating it. However, if the original data changes, the materialization process is completely repeated. Virtualization is widely used for answering consumer’s queries through virtualized access to the RDF. For each query, the RDF is regenerated, but only from the parts of the heterogeneous data sources necessary to answer the query. If these data sources change, the changes are immediately used to answer a query.

*Contribution 3: Producer and consumer involvement* I will balance the producers’ and consumers’ efforts by involving the consumer in the generation process into a new hybrid approach to divide the efforts among both producers and consumers. Example: producers may provide data to consumers to generate a part of the RDF themselves to answer their own query. The hybrid approach leverages materialization for parts which are heavily used among multiple consumers e.g. “*all departing trains in all stations in Belgium*”, while it leverages virtualization for other parts which are specifically for a single consumer e.g. “*next departing train near my location*”. Consumers can combine these parts together to answer their query. This way, queries can be answered without putting the burden on the producer only. For example, answering the query “*When does the next train depart in the nearest station?*” can use virtualized RDF to retrieve the nearest station and materialized RDF to retrieve the departing trains for the station. While the nearest station is specific for a given consumer, the list of departing trains for a station is re-usable for multiple consumers.

## 5 Evaluation Plan

This research will be evaluated through a Systematic Literature Review (SLR), a benchmark to measure the various key factors of the existing approaches, and the validation of my hybrid approach. The benchmark results are used to evaluate the proposed guidelines. Moreover, this hybrid approach will be used in several use cases such as public transport route planning and virtual assistants.

*Contribution 1: Systematic Literature Review* I executed a SLR to identify key factors, approaches and open issues of RDF generation (Section 4, Part 1).

**H1 validation:** I accept my hypothesis for RQ1, several approaches e.g. RML [6], SPARQL-Generate [11], or ShExML [14] exist for generating RDF from heterogeneous data. I identified several key factors e.g. data size, mapping rule execution, joins, and open issues e.g. applying conditions on data during the generation.

*Contribution 2: Benchmark* I will establish a benchmark based on GTFS-Madrid-Bench [22] to evaluate existing materialization and virtualization approaches against the key factors determined in the SLR to select the right approach for generating RDF. I chose to build upon the GTFS-Madrid-Bench because it already measures similar metrics, but only for virtualization approaches. I will extend this benchmark and add more metrics to cover materialization approaches. I will use the following scaling parameters: original data size, number of mapping rules, query types, and update frequency of the original data. The following metrics are inherited from the GTFS-Madrid-Bench:

- **Total execution time (s)**: Time to return the fully query answer.
- **Number of answers**: Number of answers returned.
- **RAM consumption (GB)**: Amount of memory used to answer a query.
- **Initial delay (s)**: Time to return the first part of the answer.
- **Loading time (s)**: Time for loading the ontology, mappings, and query.
- **Number of requests**: Executed number of requests.
- **Source selection time (s)**: Time for selecting all sources for an answer.
- **Results aggregation time (s)**: Time for aggregating subqueries' results.
- **Query generation time (s)**: Time for generating the query/queries.
- **Query rewriting time (s)**: Time for rewriting query into subqueries.
- **Query execution time (s)**: Time for executing the query on the sources.
- **Query translation time (s)**: Time for translating a query into a different query for a source.

I will add additional metrics to cover materialization approaches as well:

- **Selectivity**: Parts of a dataset used for answering a query.
- **Bandwidth (GB)**: Bandwidth necessary to answer a query.
- **CPU usage (%)**: CPU usage to answer a query.
- **Storage (GB)**: Storage used to store the data to answer a query.
- **Data freshness (s)**: Integration time for original data changes in the RDF.

Based on the benchmark results, I will provide producer guidelines to determine if materialization or virtualization is suitable given their own resources and how the generated RDF is consumed. I will apply and validate these guidelines on two use cases: public transport route planning and virtual assistants.

**H2 validation**: I am currently in the process of extending and setting up this benchmark. I can accept my hypothesis if I have at least one crossing point between materialization and virtualization for each key factor.

*Contribution 3: Producer & consumer involvement* I will adapt the benchmark introduced previously for my proposed hybrid approach with additional metrics, e.g., *cacheability* or *type of hardware (embedded systems, desktops, servers)*, and also metrics on the consumer side since the consumer is now involved. This way, I can compare my hybrid approach with existing materialization and virtualization approaches. These results will validate if the hybrid approach is more suitable for some use cases, e.g., public transport route planning or virtual assistants, compared to only materialization or virtualization.

## 6 Intermediate Results

This research has already a few intermediate results such as an under-review Systematic Literature Review (SLR) paper at the Journal of Web Semantics<sup>4</sup> and the paper “Leveraging Web of Things W3C recommendations for knowledge graphs generation” [33] at the ICWE 2021 conference<sup>5</sup> (published in May 2021).

In the SLR paper, I collected papers from 42 sources (workshops, journals, conferences, and digital libraries) over the past 20 years, resulting in 52 analyzed papers. This SLR confirmed that two approaches exist for generating RDF: materialization and virtualization. I discussed how these approaches differ in terms of schema transformations, data transformations, implementations and open issues. Moreover, it showed that the producer’s and consumer’s effort is unbalanced. This SLR answers RQ1, and confirms its hypothesis.

“Leveraging Web of Things W3C recommendations for knowledge graphs generation” [33], paper introduces RML’s Logical Target to specify where (parts of) the RDF must be exported, e.g., a triple store, a file, etc. RML’s Logical Target is a step towards a hybrid approach since it allows to export parts of the RDF to different targets e.g. an materialization target or an virtualization target. This way, I can export parts which are frequently re-used among multiple consumers through materialization, while other parts with a virtualization approach. For example: a public transport schedule is exported to an materialization target because it is re-used among multiple consumers while the route planning is handled by an virtualization target since routes are consumer-specific.

In the next months, I plan to develop the aforementioned benchmark that evaluates existing materialization and virtualization approaches, providing producer guidelines to select an approach given its resources and how the generated RDF is consumed. Afterwards, I will compare my proposed hybrid approach with existing approaches and investigate when a hybrid approach is more suitable than materialization or virtualization. This will be evaluated through public transport route planning and virtual assistant use cases.

## 7 Conclusion and Lessons Learned

This research already led to a better understanding of materialization and virtualization approaches for RDF generation and how they are designed to transform large amounts of data or answer specific questions.

Preliminary results of the Systematic Literature Review already highlighted open issues, key factors, and existing approaches of RDF generation. Currently, there is no way to determine which approach should be used depending on computing resources and how the RDF is consumed. Moreover, some use cases, e.g., public transport route planning or virtual assistants, need to answer multiple types of queries. A hybrid approach combining materialization and virtualization may prove to be better than existing approaches.

<sup>4</sup> <https://www.websemanticsjournal.org/>

<sup>5</sup> <https://icwe2021.webengineering.org/>

## References

1. Bansal, S., Kagemann, S.: Integrating big data: A semantic extract-transform-load framework. *Computer*, 42–50 (2015)
2. Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R.: Linking data to ontologies. In: *Journal on Data Semantics X*, pp. 133–173 (2008)
3. Das, S., Sundara, S., Cyganiak, R.: R2RML: RDB to RDF Mapping Language. Working group recommendation, World Wide Web Consortium (W3C) (2012)
4. Slepicka, J., Yin, C., Szekely, P.A., Knoblock, C.A.: Kr2rml: An alternative interpretation of r2rml for heterogeneous sources. In: *Proceedings of the 6<sup>th</sup> International Workshop on Consuming Linked Data (COLD 2015)* (2015)
5. Michel, F., Djiménou, L., Faron-Zucker, C., Montagnat, J.: Translation of Heterogeneous Databases into RDF, and Application to the Construction of a SKOS Taxonomical Reference. In: *International Conference on Web Information Systems and Technologies*, pp. 275–296 (2015)
6. Dimou, A., Vander Sande, M., Colpaert, P., Verborgh, R., Mannens, E., Van de Walle, R.: RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data. In: *Proceedings of the 7<sup>th</sup> Workshop on Linked Data on the Web* (2014)
7. Iglesias, E., Jozashoori, S., Chaves-Fraga, D., Collarana, D., Vidal, M.-E.: SDM-RDFizer: An RML Interpreter for the Efficient Creation of RDF Knowledge Graphs. In: *Proceedings of the 29<sup>th</sup> ACM International Conference on Information & Knowledge Management* (2020)
8. Chaves-Fraga, D., Ruckhaus, E., Priyatna, F., Vidal, M.-E., Corcho, O.: Enhancing virtual ontology based access over tabular data with morph-csv. *Semantic Web*, 1–34 (2021)
9. Haesendonck, G., Maroy, W., Heyvaert, P., Verborgh, R., Dimou, A.: Parallel RDF generation from heterogeneous big data. In: *Proceedings of the International Workshop on Semantic Big Data - SBD '19* (2019)
10. Harris, S., Seaborne, A.: SPARQL 1.1 Query Language. Recommendation, World Wide Web Consortium (W3C) (2013)
11. Lefrançois, M., Zimmermann, A., Bakerally, N.: A SPARQL extension for generating RDF from heterogeneous formats. In: *The Semantic Web 14<sup>th</sup> International Conference, ESWC 2017, Portorož, Slovenia, May 28 – June 1, 2017, Proceedings*, pp. 35–50 (2017)
12. Daga, E., Asprino, L., Mulholland, P., Gangemi, A.: Facade-X: An Opinionated Approach to SPARQL Anything. In: *Further with Knowledge Graphs – Proceedings of the 17<sup>th</sup> International Conference on Semantic Systems, 6–9 September 2021, Amsterdam, The Netherlands*, pp. 58–73 (2021)
13. Bischof, S., Decker, S., Krennwallner, T., Lopes, N., Polleres, A.: Mapping between RDF and XML with XSPARQL. *Journal on Data Semantics* (3), 147–185 (2012)
14. García-González, H., Boneva, I., Staworko, S., Labra-Gayo, J.E., Lovelle, J.M.C.: ShExML: improving the usability of heterogeneous data mapping languages for first-time users. *PeerJ Computer Science*, 318 (2020)
15. Prud'hommeaux, E.: Shape Expressions 1.0 Primer. Member submission, World Wide Web Consortium (W3C) (2014)
16. Priyatna, F., Corcho, O., Sequeda, J.: Formalisation and experiences of R2RML-based SPARQL to SQL query translation using morph. In: *Proceedings of the 23<sup>rd</sup> International Conference on World Wide web*, pp. 479–490 (2014)

17. Sequeda, J.F., Miranker, D.P.: Ultrawrap: SPARQL execution on relational data. *Journal of Web Semantics*, 19–39 (2013)
18. Calvanese, D., Cogrel, B., Komla-Ebri, S., Kontchakov, R., Lanti, D., Rezk, M., Rodriguez-Muro, M., Xiao, G.: Ontop: Answering SPARQL Queries over Relational Databases. *Semantic Web Journal* (3), 471–487 (2017)
19. Mami, M.N., Graux, D., Scerri, S., Jabeen, H., Auer, S., Lehmann, J.: Squerall: Virtual ontology-based access to heterogeneous and large data sources. *Lecture Notes in Computer Science*, pp. 229–245. Springer (2019)
20. Endris, K.M., Rohde, P.D., Vidal, M.-E., Auer, S.: Ontario: Federated Query Processing Against a Semantic Data Lake. In: *Database and Expert Systems Applications: 30<sup>th</sup> International Conference, DEXA, Part I*, pp. 379–395 (2019)
21. Khan, Y., Zimmermann, A., Jha, A., Gadepally, V., D’Aquin, M., Sahay, R.: One Size Does Not Fit All: Querying Web Polystores. *IEEE Access*, 9598–9617 (2019)
22. Chaves-Fraga, D., Priyatna, F., Cimmino, A., Toledo, J., Ruckhaus, E., Corcho, O.: Gtfs-madrid-bench: A benchmark for virtual knowledge graph access in the transport domain. *Journal of Web Semantics*, 100596 (2020)
23. Bizer, C., Schultz, A.: The Berlin SPARQL benchmark. *International Journal on Semantic Web and Information Systems (IJSWIS)* (2), 1–24 (2009)
24. Guo, Y., Pan, Z., Heflin, J.: Lubm: A benchmark for owl knowledge base systems. *Journal of Web Semantics* (2), 158–182 (2005). Selected Papers from the International Semantic Web Conference, 2004
25. Schmidt, M., Hornung, T., Lausen, G., Pinkel, C.: Sp<sup>2</sup>bench: a sparql performance benchmark. In: *2009 IEEE 25th International Conference on Data Engineering*, pp. 222–233 (2009)
26. Hasnain, A., Mehmood, Q., Sana e Zainab, S., Saleem, M., Warren, C., Zehra, D., Decker, S., Rebholz-Schuhmann, D.: Biofed: federated query processing over life sciences linked open data. *Journal of biomedical semantics* (1), 1–19 (2017)
27. Morsey, M., Lehmann, J., Auer, S., Ngonga Ngomo, A.-C.: Dbpedia sparql benchmark – performance assessment with real queries on real data. In: *The Semantic Web – ISWC 2011*, pp. 454–469 (2011)
28. Rivero, C.R., Schultz, A., Bizer, C., Ruiz Cortés, D.: Benchmarking the performance of linked data translation systems. In: *LDOW 2012: WWW2012 Workshop on Linked Data on the Web (2012)*, (2012)
29. Lanti, D., Rezk, M., Xiao, G., Calvanese, D.: The npd benchmark: Reality check for obda systems. In: *EDBT*, pp. 617–628 (2015)
30. Chaves-Fraga, D., Endris, K.M., Iglesias, E., Corcho, O., Vidal, M.-E.: What are the parameters that affect the construction of a knowledge graph? In: Panetto, H., Debruyne, C., Hepp, M., Lewis, D., Ardagna, C.A., Meersman, R. (eds.) *On the Move to Meaningful Internet Systems: OTM 2019 Conferences*, pp. 695–713 (2019)
31. Verborgh, R., Vander Sande, M., Hartig, O., Van Herwegen, J., De Vocht, L., De Meester, B., Haesendonck, G., Colpaert, P.: Triple pattern fragments: A low-cost knowledge graph interface for the web. *Journal of Web Semantics*, 184–206 (2016)
32. Machado, G.V., Cunha, Í.S., Pereira, A.M., Oliveira, L.B.: Dod-etl: distributed on-demand etl for near real-time business intelligence. *Journal of Internet Services and Applications*, 1–15 (2019)
33. Van Assche, D., Haesendonck, G., De Mulder, G., Delva, T., Heyvaert, P., De Meester, B., Dimou, A.: Leveraging Web of Things W3C Recommendations for Knowledge Graphs Generation. In: *Web Engineering. Lecture Notes in Computer Science*, pp. 337–352. Springer (2021)