FDA-DBRepo: A Data Preservation Repository Supporting FAIR Principles, Data Versioning and Reproducible Queries

Martin Weise
TU Wien
Austria
martin.weise@tuwien.ac.at
0000-0003-4216-302X

Cornelia Michlits
TU Wien
Austria
cornelia.michlits@tuwien.ac.at
0000-0002-7810-0314

Moritz Staudinger
TU Wien
Austria
moritz.staudinger@tuwien.ac.at
0000-0002-5164-2690

Eva Gergely
Universität Wien
Austria
eva.gergely@univie.ac.at
0000-0003-4218-5990

Kirill Stytsenko
Universität Wien
Austria
kirill.stytsenko@univie.ac.at
0000-0003-0884-3937

Raman Ganguly
Universität Wien
Austria
raman.ganguly@univie.ac.at
0000-0002-9837-0047

Andreas Rauber
TU Wien
Austria
andreas.rauber@tuwien.ac.at
0000-0002-9272-6225

Database preservation frequently happens post-factum: databases are transferred and migrated into preservation formats and environments after a project has ended. This increases the risks concerning incompatibility and pushes the preservation burden after the initial lifetime and use of the data. We propose a database repository infrastructure, where databases are created, used and preserved directly in the data curation environment. This increases the FAIRness of the data curated as professional data stewardship activities accompany the databases right from the onset. We present the FAIR Data Austria Database Repository (FDA-DBRepo) infrastructure and provide a first version of an open-source reference implementation.

Keywords – Database Preservation, Research Data, FAIR Data, Data Citation, Reproducibility

Conference Topic – Exploring New Horizons

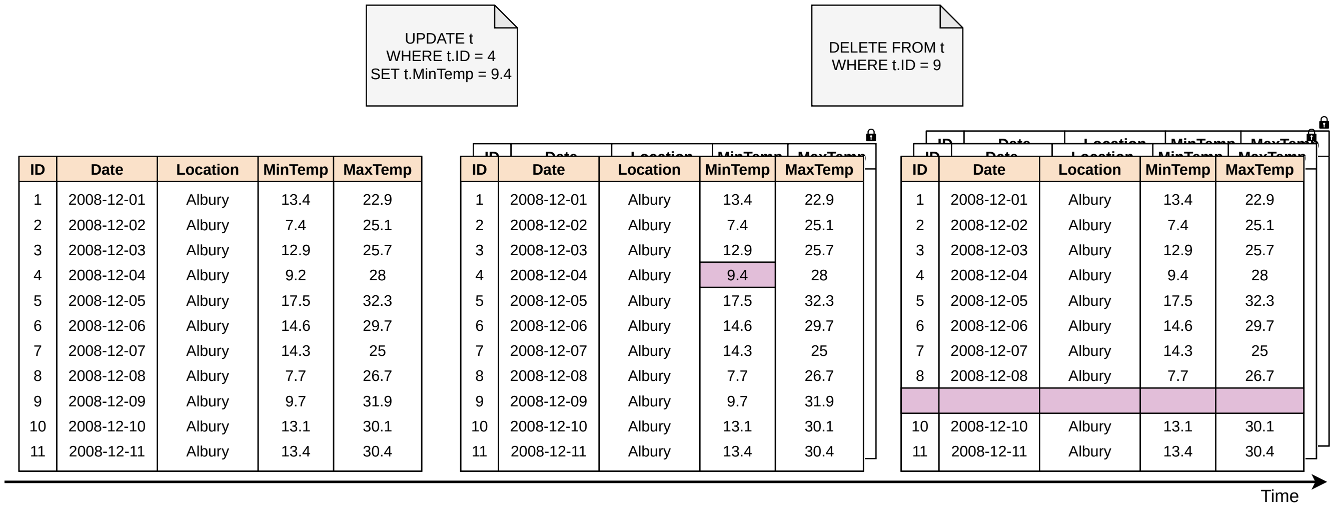
I. Motivation

Relational databases constitute a core resource in virtually all domains, with a significant part of its value being derived from long-term availability of such data. In an increasing number of settings we thus face the need to preserve data stored in relational database management systems (RDBMS) and make it accessible to third parties for research purposes or to assist with specific analytical tasks. While a lot of valuable enhancements to data (provenance, finding aids) as well as potential re-use of data happen during the life time of a project, many institutions still commence preservation activities only at its end, following frequently proliferated data life cycle models which usually have “preserve” towards the end of a cycle and often motivated solely by having to meet funding agreements or publisher requirements. This approach leads to several disadvantages: valuable expertise on data semantics is lost as project team members move on, the FAIRness principles receive too little attention as researchers in various domains lack the expertise, training and interest in data stewardship and – last but not least – the burden of setting up and operating the database frequently rests with the research team rather than with data curators and IT experts. We thus propose to explore new horizons by ensuring that research databases are designed, created and used in a proper environment, with a clear separation of concerns: researchers should focus on their core domain expertise in working with domain-specific data, while data stewards take care of data curation activities, with IT staff managing core database operations, ensuring IT security and providing appropriate interfaces and scalability.

The FAIR Data Austria (FDA) project is a part of the Cluster Research Data (“Cluster Forschungsdaten”) funded by the Austrian Federal Ministry of Education, Science and Research (BMBWF), which aims to provide infrastructure and services for knowledge transfer between universities, industry and society to support the sustainable implementation of the European Open Science Cloud (EOSC). Among other activities, three repositories will be developed and deployed as part of FDA,

17th International Conference on Digital Preservation
iPRES 2021, Beijing, China.

DOI: 10.1145/nnnnnnn.nnnnnnn



```
UPDATE t
WHERE t.ID = 4
SET t.MinTemp = 9.4
```

```
DELETE FROM t
WHERE t.ID = 9
```

ID	Date	Location	MinTemp	MaxTemp
1	2008-12-01	Albury	13.4	22.9
2	2008-12-02	Albury	7.4	25.1
3	2008-12-03	Albury	12.9	25.7
4	2008-12-04	Albury	9.2	28
5	2008-12-05	Albury	17.5	32.3
6	2008-12-06	Albury	14.6	29.7
7	2008-12-07	Albury	14.3	25
8	2008-12-08	Albury	7.7	26.7
9	2008-12-09	Albury	9.7	31.9
10	2008-12-10	Albury	13.1	30.1
11	2008-12-11	Albury	13.4	30.4

ID	Date	Location	MinTemp	MaxTemp
1	2008-12-01	Albury	13.4	22.9
2	2008-12-02	Albury	7.4	25.1
3	2008-12-03	Albury	12.9	25.7
4	2008-12-04	Albury	9.4	28
5	2008-12-05	Albury	17.5	32.3
6	2008-12-06	Albury	14.6	29.7
7	2008-12-07	Albury	14.3	25
8	2008-12-08	Albury	7.7	26.7
9	2008-12-09	Albury	9.7	31.9
10	2008-12-10	Albury	13.1	30.1
11	2008-12-11	Albury	13.4	30.4

ID	Date	Location	MinTemp	MaxTemp
1	2008-12-01	Albury	13.4	22.9
2	2008-12-02	Albury	7.4	25.1
3	2008-12-03	Albury	12.9	25.7
4	2008-12-04	Albury	9.4	28
5	2008-12-05	Albury	17.5	32.3
6	2008-12-06	Albury	14.6	29.7
7	2008-12-07	Albury	14.3	25
8	2008-12-08	Albury	7.7	26.7
10	2008-12-10	Albury	13.1	30.1
11	2008-12-11	Albury	13.4	30.4

Figure 1: Data versioning using temporal tables.

namely a file-based repository (based on Invenio[1]), a Source-Code Repository (based on Git[2]), as well as a novel database repository (FDA-DBRepo). This database repository shall support the flexible creation, use and curation of relational databases in a private-cloud hosted environment. Researchers shall be able to find and access these databases (for public databases, potentially after an embargo period), being able to execute reproducible view-queries against the data (for open databases) supporting data citation and re-use. FDA-DBRepo strives to support the FAIR Principles (Findable, Accessible, Interoperability, Reusable) [1] to the largest extent possible by including curational activities right from the onset of database creation, e.g. by adding ontology-based metadata mapping to increase findability, continuous data versioning (using temporal tables) to support reproducible queries assisting with accessibility and re-use both via web interfaces as well as APIs designed for machine interaction. Development started in the summer of 2020 with a first Open Source release foreseen by September 2021. [3]

While the approach presented does not, in itself, solve the preservation challenge for databases per-se, it ensures that such databases can be created and used directly in a repository context. This allows preservation actions to be planned and deployed from the onset during the entire life-time of the respective database. It also supports a clear separation of concerns, with the researchers being able to focus on data processing as required by their research activities, with system administrators taking care of the actual database administration, while curators are able to concentrate on data curation activities, with specific tool support being provided for all roles.

The remainder of this paper is structured as follows: Section II briefly points to some pertinent related work, followed by a presentation of technical architecture in Section III. We further discuss the current deployment of the initial prototype in Section IV and provide an outlook on future work in Section V.

II. Related Work

Database preservation is a massive challenge. Approaches usually range from archiving only the data to developing specific emulators [2] to ensure functionality can be provided in the future. Frequently, once preserved, an interested third party cannot interact with historical records directly via machine-readable interfaces anymore but has to rely on human support to access them. Data versioning on these monolithic records is limited and implies a large administrative overhead, making it viable only for rare data exports or single post-project deposits. To overcome some of these challenges, the Swiss Federal Archives developed the open format SIARD [3] to preserve relational databases in the long-term, removing the risks associated with proprietary RDBMS systems.

Anderson et al. [4] described a strategy to preserve a veterinary medicine domain scientific database using the SIARD 2 format, while still being able to query it from the organization's content management systems. Interestingly, their preservation activity would not be possible in the SIARD format. They also report difficulties using the legacy SIARD format (lack of relations, opening the file during testing, lack of functionality).

Our approach complements the state of the art archival of databases by ensuring that the databases are created centrally on dedicated hardware (to ensure their availability), curated already during the lifetime of a project (instead of long-term archival) and let individuals run reproducible view-queries on a specific data set version. It will, of course, need to incorporate export functionality to the standardized SIARD format for long-term archival of retired databases relying e.g. on the powerful Database Preservation Toolkit [5].

[1] `https://inveniordm.docs.cern.ch/`
[2] `https://git-scm.com/`
[3] `https://github.com/fair-data-austria/dbrepo`

III. Architectural Design

A. *Requirements*

The key requirements set out for the project include (1) allowing non-expert users to create and operate databases for their research purposes, (2) providing professional data management and curation support centrally from the on-set, i.e. while the data is in active use, (3) guaranteeing high scalability and flexibility, (4) supporting the FAIR principles with a specific focus on data citation, and (5) ensuring that these services can be offered in-house in a private cloud setting to ensure compliance with data protection regulations.

The creation, operation and curation of databases typically affects scientist from non computer science areas with strongly differing SQL knowledge. To drastically decrease the entry barrier for non-expert users, we provide three methods of ingesting data into the system: (1) novice users upload their data in *CSV* format and make use of the Analyze Service to propose data types as well as the primary key column, (2) apprentice users may specify the database schema directly (3) expert users can use SQL statement to ingest data into the database, (4) machines will be able to populate a table directly via a REST and AMQP interfaces, allowing e.g. a sensor to automatically insert data. These and similar use-case requirements were elicited in an initial requirements workshop with researchers from different faculties within TU Wien.

B. *Data Model*

In order to guarantee that databases are independent and can be managed autonomously, each database is encapsulated in a Docker[4] container together with required packages and libraries. Moreover, this design makes it possible to support different database versions as well as various engines, hence FDA-DBRepo is flexible and support long-term data preservation of structured research data. Since each database is assigned to a container, they can be distributed across multiple servers, which reinforces scalability. A major role is played by the Metadata Database (MDB), which contains metadata of the user databases running in the repository, e.g. table names, column names, SI units and mandatory metadata properties of research data described by Ammann et al. [6]. The MDB is the key concept and central mechanism for data stewardship in order to increase the FAIRness [1] of data. It provides a central registry of all databases deployed in the infrastructure, including additional metadata as provided by the user, the schema and data type definitions as well as basic statistical properties such as the value ranges, mean and median values per attribute, update frequencies.

C. *Software Architecture*

Microservices form the foundation for the functionalities in FDA-DBRepo and enable targeted scalability, usage of diverse technologies as well as load balancing. We describe these services in the following.

Container Service & Database Service in charge of database creation within a container in three different ways: via *CSV* upload, form or *SQL* commands, and their maintenance.

Table Service similar to the Database Service, this service supports three possibilities to create a table within a database. Depending on the chosen variant the Table Service automatically generates valid *SQL* statements.

Analyze Service maps the metadata about databases and tables immediately after creation to controlled vocabulary, forwards the metadata to the MDB and makes suggestion on datatypes for columns within a *CSV* file as well as primary key recommendations.

Query Service is responsible for storing queries in normalized form, assigning a persistent identifier (PID) to each query as well as executing and re-executing queries against a timestamp in order to guarantee reproducibility.

Search Service enables user to search in the MDB in order to make databases findable.

Briefly recapped, the Container-, Database- and Table Services ensure the core user database operations within FDA-DBRepo. The MDB, Analyze and Search Services enable FAIRness. Data versioning and PIDs created within the Query Service assure that identical results are retrieved on re-execution, hence even fine-granular subsets of data are citable. An overview of the system including the listed microservices is provided in Figure 2.

D. *Data Curation Services*

As data preservation and data curation become increasingly important, especially on evolving data collections, our proposed repository infrastructure aims at providing these services centrally to relieve the researchers from data management activities that are beyond their core focus of interest and expertise. Specifically, to increase the FAIRness of our data, enhanced metadata mapping services are planned to assist with the use of or mapping to controlled vocabularies for attribute names using domain-specific data dictionaries (e.g. the INSPIRE registries[5] or the Data Type Registry[6], as well as assigning appropriate measurement units via the International System of Quantities (ISO 80000) and International System of Units (SI) [7]. By semi-automatically, i.e. suggesting to the user a mapping of attribute names to controlled vocabularies as

[4] https://www.docker.com/

[5] http://inspire.ec.europa.eu/registry

[6] http://typeregistry.org/registrar/

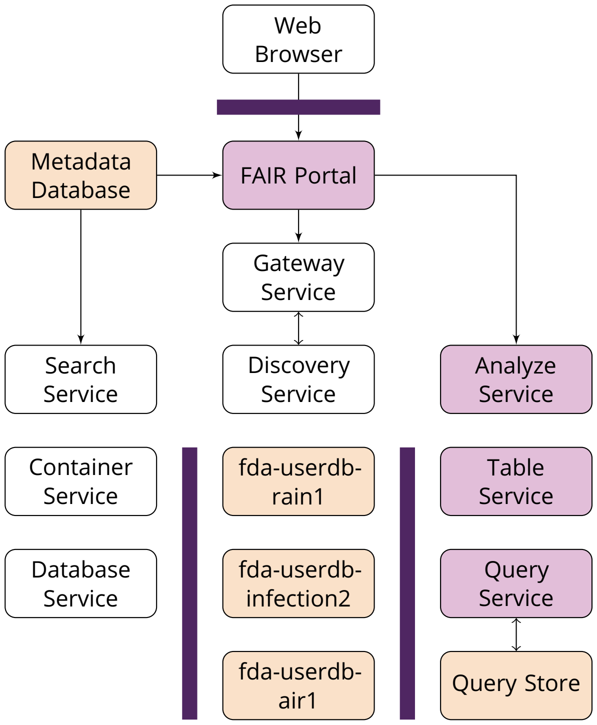


Figure 2: Overview architecture of the FDA-DBRepo infrastructure with three exemplary user databases protected by a firewall barrier around them.

well as assigning associated measurement units when creating the database (or ex-post for already existing DBs) provides a rich basis for finding data. It allows a user, for example, to find all databases that contain <*temperature*> (irrespective of whether the attribute is named “temp”, “temperature” or “温度”) measurements in the *range* of *273.15 - 373.15 degrees Kelvin* or specific *min/max* values, all irrespective of the actual measurement unit used for storing or querying the data due to the possibility of automatic conversion between these.

A second focus is on support for citing arbitrary subsets of the datasets even when the data is evolving over time as many scientific journals and conferences already require data to be appropriately cited and provided along with the manuscript as supplementary material and it constitutes a core requirement for reproducibility of scientific research relying on data. Each database hosted in FDA-DBRepo is automatically versioned, by adding a creation and deletion timestamp per tuple. With this it is possible to reconstruct the exact data at any given point in time, without the need of freezing the complete dataset upon each change. Additionally, queries executed are stored and assigned a persistent identifier following the guidelines provided by the RDA Working Group on Data Citation [8], [9]. Our approach creates a citeable data set with every data update using temporal tables (native data versioning capability of many RDBMS’). Figure 1 shows an update operation on a table[7] with temporal tables extension, creating a new version; a subsequent deletion of a row creates a third version of the table that can be cited. Along with the Query Store, the data set citation allows for a re-execution of stored queries and comparing the results.

Upon retirement of the database, support for export as SIARD archive to a long-term block device using the SIARD toolkit [5] is currently under investigation. This process is not yet fully automated within the system and requires human interaction.

IV. Deployment

Currently, we deployed a development prototype of FDA-DBRepo on a dedicated server within TU Wien without access to other servers in the same network. It is equipped with 16 vCPUs (AMD EPYC 7702 $@2.0$ GHz), 32 GB RAM, 100 GB SSD persistent storage and runs the Docker Engine v20.10 along with Docker Compose v1.29.

The initial FAIR Portal prototype of our prototype as depicted in Figure 4 addresses the basic use-cases: users are able to create, populate and query databases. They can find a database by searching the metadata via table description and attributes, with searches via controlled vocabularies, by measurement units or value ranges being under development. Querying a specific database returns paged results via the web interface, supporting downloads of entire result set as *CSV* file or by connecting directly to an API.

To ensure scalability, we use multiple instances of PostgreSQL[8] running inside individual Docker containers. We decided against full virtualization in favor of lightweight, containerized services and strong scalability capabilities (e.g. by increasing the number of replicas). Docker also greatly reduces configuration needed to run multiple databases at once, listening to different ports and within different virtual networks. To overcome limitations of interfacing with any particular RDBMS system we decided to use Hibernate [10] as it supports a multitude of RDBMS’ and abstracts the operations, only depending on appropriate mappings to create a session that manages the user containers.

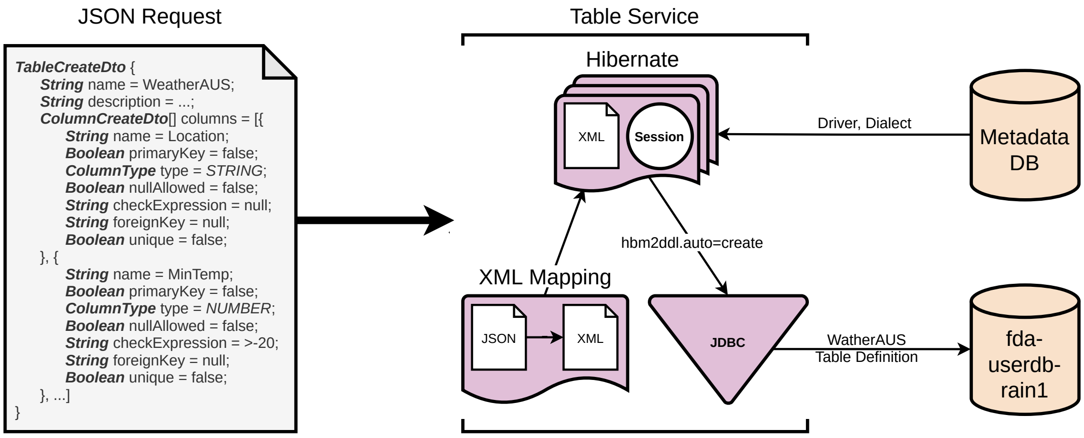
V. Conclusions & Future Work

We presented our data preservation repository (FDA-DBRepo) supporting FAIR principles that allows data stewards to curate data already early on during projects rather than having to start addressing preservation challenges after the end of a project. We discussed the open-source prototype implementation that already supports the key features (data ingest, -versioning, -citation, based on expertise of the user and the Metadata Database) required by researchers, allowing domain experts to focus on their domain-specific interaction with data while data stewards and IT experts take care of data curation and IT operations.

While several key features (e.g. controlled vocabu-

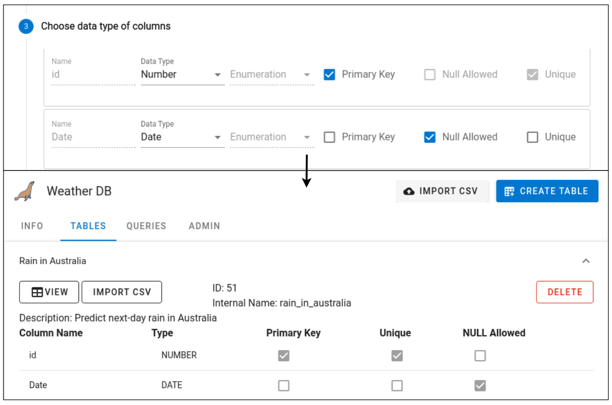
[7] `https://www.kaggle.com/jsphyg/weather-dataset-rattle-package`

[8] `https://www.postgresql.org/`



```
TableCreateDto {
    String name = WeatherAUS;
    String description = ...;
    ColumnCreateDto[] columns = [{
        String name = Location;
        Boolean primaryKey = false;
        ColumnType type = STRING;
        Boolean nullAllowed = false;
        String checkExpression = null;
        String foreignKey = null;
        Boolean unique = false;
    }, {
        String name = MinTemp;
        Boolean primaryKey = false;
        ColumnType type = NUMBER;
        Boolean nullAllowed = false;
        String checkExpression = >-20;
        String foreignKey = null;
        Boolean unique = false;
    }, ...]
}
```

Figure 3: FDA-DBRepo abstracts table operations to a database engine-independent Table Service capable of handling create, read, update and delete operations.



Name	Data Type	Enumeration			
id	Number		☑ Primary Key	□ Null Allowed	☑ Unique
Date	Date		□ Primary Key	☑ Null Allowed	□ Unique

Rain in Australia

ID: 51
Internal Name: rain_in_australia

Description: Predict next-day rain in Australia

Column Name	Type	Primary Key	Unique	NULL Allowed
id	NUMBER	☑	☑	□
Date	DATE	□	□	☑

Figure 4: Table creation for novice and apprentice users

laries for data type mapping, export to SIARD) are still under development, a range of additional requirements are being collected through interaction with researchers. These include real-time feeding of data using stream processing technology (e.g. Apache Storm[9]), support for other types of database engines (e.g. NoSQL, graph databases) or support for queries spanning across several databases to allow record linking.

Acknowledgment

Part of this work is supported by the Ministry of Education, Science and Research under the program for Digital and Social Transformation in Higher Education.

References

[1] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, *et al.*, "The FAIR Guiding Principles for Scientific Data Management and Stewardship," *Scientific Data*, vol. 3, no. 1, pp. 1–9, 2016.

[2] S. Granger, "Emulation as a Digital Preservation Strategy," *D-Lib Magazine*, vol. 6, no. 10, 2000. doi: 10.1045/october2000-granger.

[3] H. Bruggisser, G. Büchler, A. Dubois, M. Kaiser, L. Kansy, M. Lischer, C. Röthlisberger-Jourdan, H. Thomas, and A. Voss, *Siard-formatspezifikation*, Verein eCH, Zürich, Switzerland, 2013.

[4] B. Anderson, S. Braxton, H. Imker, and T. Popp, "The Art of Preserving Scientific Data: Building Collaboration into the Preservation of a Legacy Database," in *iPRES 2018 15th Intl. Conf. on Digital Preservation*, Boston, MA, USA, 2018.

[5] B. Ferreira, L. Faria, J. C. Ramalho, and M. Ferreira, "Database Preservation Toolkit: A Relational Database Conversion and Normalization Tool," in *iPRES 2016 13th Intl. Conf. on Digital Preservation*, CCSDS 650.0-M-2, Bern, Switzerland, 2016.

[6] DataCite Metadata Working Group, "DataCite Metadata Schema for the Publication and Citation of Research Data," 2016. doi: 10.5438/0012.

[7] A. Thompson and B. N. Taylor, *Guide for the Use of the Intl. System of Units. National Institute of Standards and Technology*, 811. National Institute of Standards and Technology, 2008. doi: 10.6028/NIST.SP.811e2008.

[8] A. Rauber, A. Asmi, D. van Uytvanck, and S. Pröll, *Data Citation of Evolving Data: Recommendations of the Working Group on Data Citation (WGDC)*, Oct. 2015. doi: 10.15497/RDA00016. [Online]. Available: https://doi.org/10.15497/RDA00016.

[9] A. Rauber, A. Asmi, D. van Uytvanck, and S. Proell, "Identification of Reproducible Subsets for Data Citation, Sharing and Re-Use," *Bulletin of the IEEE Tech. Comm. on Digital Libraries (TCDL)*, vol. 12, no. 1, 2016. doi: 10.5281/zenodo.4048304.

[10] E. J. O'Neil, "Object/Relational Mapping 2008: Hibernate and the Entity Data Model," in *Proceedings of the 2008 ACM SIGMOD Intl. Conf. on Management of Data*, ser. SIGMOD '08, Vancouver, Canada: Association for Computing Machinery, 2008, 1351–1356. doi: 10.1145/1376616.1376773.

[9] https://storm.apache.org