

FAIR Data Austria Database Repository

Martin Weise, Moritz Staudinger, Cornelia Michlits

Vienna University of Technology

June 22, 2021

Database Repository

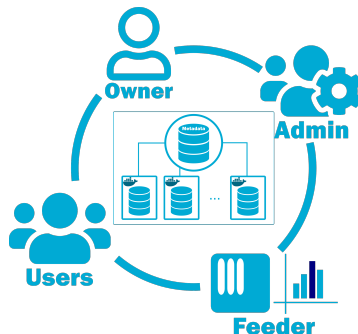
Some problems of currently used database repositories

- Databases are set up locally at research units
- Database **admin skills** required
- Local usage does not motivate to provide metadata
- Lack of **metadata** yield to unusable DB dumps after end of a project
- **Unversioned** data causes lack of reproducibility

Database Repository

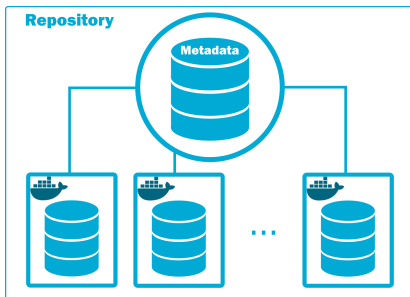
Goal of the FAIR Data Repository:

- Private cloud hosted repository
- Database management outsourced to repository infrastructure
- **Metadata** makes databases searchable and reusable
- Data **versioning** with timestamps guarantees reproducibility
- Data is **cite-able**
- Support different levels of SQL-knowledge

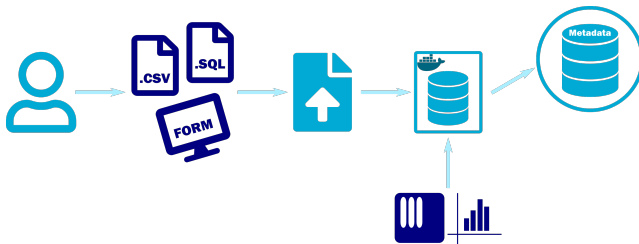


Set-up Overview

- Each database is encapsulated in a docker container
- Metadata-Database contains relevant information to make databases findable and cite-able
- Microservice Architecture used



Workflow Example



- Upload CSV, SQL dump
- Database creation via form
- Feeding database via REST interface

A FAIR Database Repository

The Metadata-Database makes the repository FAIR.

Findable & Reusable: MDB contains data about

- ... the **data**, e.g., creator, publisher, data license, publication year, resource type, data description
- ... the **database schema**, e.g., database name, table names, attribute names,
- ... **measurement units** of numerical columns
- ... basic **statistical** information about the data.

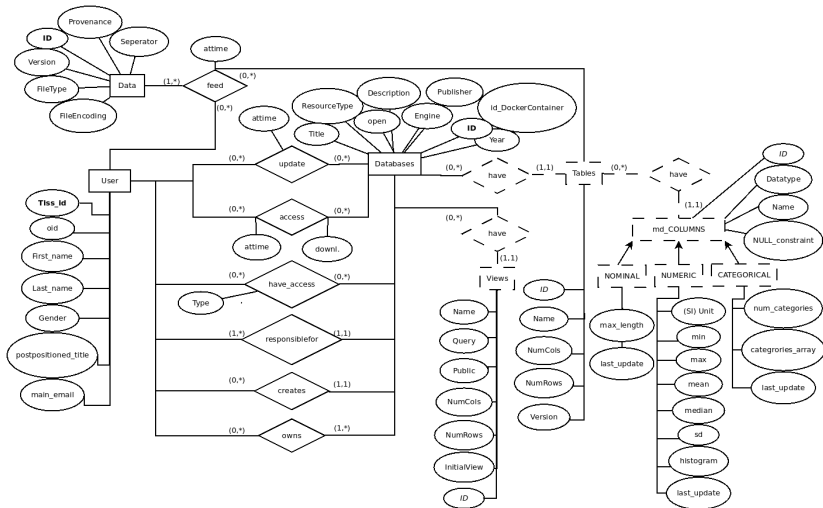
A FAIR Database Repository

The Metadata-Database makes the repository FAIR.

Accessible & Interoperable:

- Data mapped **controlled vocabularies**
- Find DB via search on the metadata
- Use webinterface (SQL, faceted browsing) to query DB
- Query DB via REST interface
- Dynamic data citation

Metadata-Database



Microservice Architecture

Microservices *allows us to make our system modular and scalable.*

- Each Services is encapsulated in a docker container
- Interact with REST requests
- Services for all different tasks:
 - Container Management
 - Database Management
 - Query Execution
 - ...

Data Versioning

Data Versioning *allows to track all changes in a database*

- Additional Creation/Deletion Timestamp per Table
- Update is handled, with new Entry in Table

id	firstname	lastname	created	deleted
1	max	mustermann	2021-05-05 12:00:00	2021-05-06 16:00:00
2	Martin	Weise	2021-05-05 12:00:00	null
1	Max	Mustermann	2021-05-06 16:00:00	null
3	Moritz	Staudinger	2021-05-06 16:00:00	null
4	Cornelia	Michlits	2021-05-06 16:00:00	null

Query Store & Execution

The Query Store saves all executed queries for re-execution and citation.

- On Execution, each query is **saved** in Querystore
- Queries are **normalized**, so that identical queries are recognizable
- Each query is assigned a **persistent identifier**
- Assures Re-execution with **identical result**
- Re-execution allows subsets to be identified and cited
- Hashing ensures correctness of results.