Approaches to Making Data Citeable

Recommendations of the RDA Working Group

Andreas Rauber,
Ari Asmi, Dieter van Uytvanck
Stefan Pröll

RESEARCH DATA ALLIANCE
United we stand

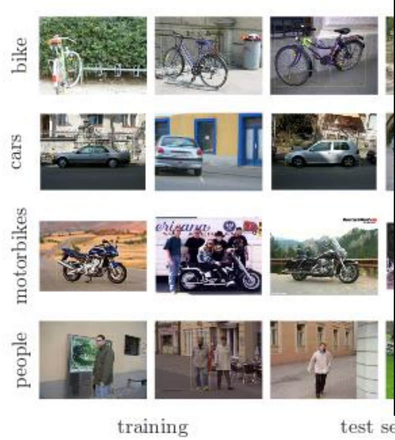
FACULTY OF !NFORMATICS

Outline

- Challenges addressed by the WG
- Recommendation of the RDA Working Group
- Benefits
- Summary

Data Citation

- Citing data may seem easy
  - from providing a URL in a footnote
  - via providing a reference in the bibliography section
  - to assigning a PID (DOI, ARK, …) to dataset in a repository
- What's the problem?



~~Fig. 2. Image examples with ground truth object annotation for different categories of~~ he PASCAL 2005 challenge. The dataset may be obtained from http://www.pascal-network.org/challenges/VOC.

present suggests the presence of 4 lineages, if *Eriophorum crinigerum* groups with the rest of Scirpeae: the Dulichieae (*Dulichium* + *Blysmus*), *Khaosokia*, Scirpeae, and Cariceae. More targeted work on the Scirpeae will be necessary to clarify this. The Fuireneae+Cypereae clade presents a similar problem: the monophyletic Cypereae contains 2 well-supported clades (*Cyperus* s.l. and *Ficinia/Isolepis*), but the taxa usually attributed to the Fuireneae form a polytomy below Cypereae (Figs. 6 and 7). Previous studies have seen these lineages positioned in many different locations, usually without strong support (Simpson et al. 2007; Muasya et al. 2009), but a study using *ndhF* and *psbB-psbH* (Hinchliff et al. 2010) showed strong support for a Fuireneae grade leading to the Cypereae. Additional sampling of *ndhF* and other data-rich cpDNA regions such as *psbB-psbH* and perhaps *matK* may help clarify these relationships.

Overall, 9 clades are strongly supported and morphologically diagnosible (Mapanioideae, Trilepideae, Sclerieae, Schoeneae, Rhynchosporeae, Abildgaardieae, *Eleocharis*, and Cypereae), and should be recognized in a new classification, as previous classifications are clearly do not define phylogenetic lineages as we now know them. Additional research will clarify how many diagnosible lineages will need to be recognized within the *Carex* + Dulichieae + *Khaosokia* + Scirpeae clade and the Fuireneae assemblage.

SUPPLEMENTARY MATERIAL

Data files and/or other supplementary information related to this paper have been deposited on Dryad at http://datadryad.org under doi: 10.5061/dryad.6p76c3pb.

FUNDING

This work was supported by the National Science Foundation [DEB 1011206 to C.E.H.].

ACKNOWLEDGEMENTS

Useful discussions from WSU/UI PuRGe were invaluable, as was specific feedback from Matt Pennell, Luke Harmon, and Jeremiah Busch. Many thanks to Steve Orzell and Edwin Bridges of Avon Park, FL, for companionship and hospitality in the field that mistakenly went unobserved in an earlier paper.

REFERENCES

Aberer A.J., Krompass D., Stamatakis A. 2012. Pruning rogue taxa improves phylogenetic accuracy: an efficient algorithm and webservice. Systematic Biology (in press) doi:10.1093/sysbio/sys078.

…ntitative analysis of the problem of constructing an NLI we …ows: we downloaded a dataset which has been frequently used …n of natural language interfaces, i.e. the Geobase dataset col… and his students[1]. The Geobase dataset describes states, cities, rivers and roads in the U.S., together with attributes such as …, population (state, city), length (river), height (mountain, lo…

…consists of a set of 880 test questions (actually 883 questions) … through a web interface hosted at the University of Austin in … the 883 test questions for our analysis. After downloading the …g), we converted the whole dataset into the ontology languages …WL[3]. The datasets are available from http://www.cimiano.de …atasets and other Material → ORAKEL.

[1] This dataset is available from: http://www.cs.utexas.edu/users/ml/nldata.html
[2] There is also a dataset consisting of 250 questions available from the University of Texas but this is merely a subset of the larger dataset.
[3] http://www.w3.org/TR/owl-features/

ifs FACULTY OF !NFORMATICS

RDA RESEARCH DATA ALLIANCE United we stand

Citation of Dynamic Data

- Citable datasets have to be static
  - Fixed set of data, no changes:
    no corrections to errors, no new data being added
- But: (research) data is **dynamic**
  - Adding new data, correcting errors, enhancing data quality, …
  - Changes sometimes highly dynamic, at irregular intervals
- Current approaches
  - Identifying entire data stream, without any versioning
  - Using “accessed at” date
  - “Artificial” versioning by identifying batches of data (e.g. annual), aggregating changes into releases (time-delayed!)
- Would like to cite precisely the **data as it existed at certain point in time**, without delaying release of new data

Granularity of Data Citation

- What about the **granularity** of data to be cited?
  - Databases collect enormous amounts of data over time
  - Researchers use specific subsets of data
  - Need to identify precisely the subset used
- Current approaches
  - Storing a copy of subset as used in study -> scalability
  - Citing entire dataset, providing textual description of subset -> imprecise (ambiguity)
  - Storing list of record identifiers in subset -> scalability, not for arbitrary subsets (e.g. when not entire record selected)
- Would like to be able to cite precisely the **subset of (dynamic) data used** in a study

Outline

- Joint Declaration of Data Citation Principles
- Challenges in non-trivial settings
- Recommendation of the RDA Working Group
- Benefits
- Summary

Making Dynamic Data Citeable

Data Citation: Data + Means-of-access

- Data → time-stamped & versioned (aka history)

Researcher creates working-set via some interface:

- Access → **assign PID to QUERY**, enhanced with
  - **Time-stamping** for re-execution against versioned DB
  - **Re-writing** for normalization, unique-sort, mapping to history
  - **Hashing** result-set: verifying identity/correctness

  leading to landing page

S. Pröll, A. Rauber. **Scalable Data Citation in Dynamic Large Databases: Model and Reference Implementation.** In IEEE Intl. Conf. on Big Data 2013 (IEEE BigData2013), 2013
http://www.ifs.tuwien.ac.at/~andi/publications/pdf/pro_ieeebigdata13.pdf

Data Citation – Deployment

Note: query string provides excellent provenance information on the data set!

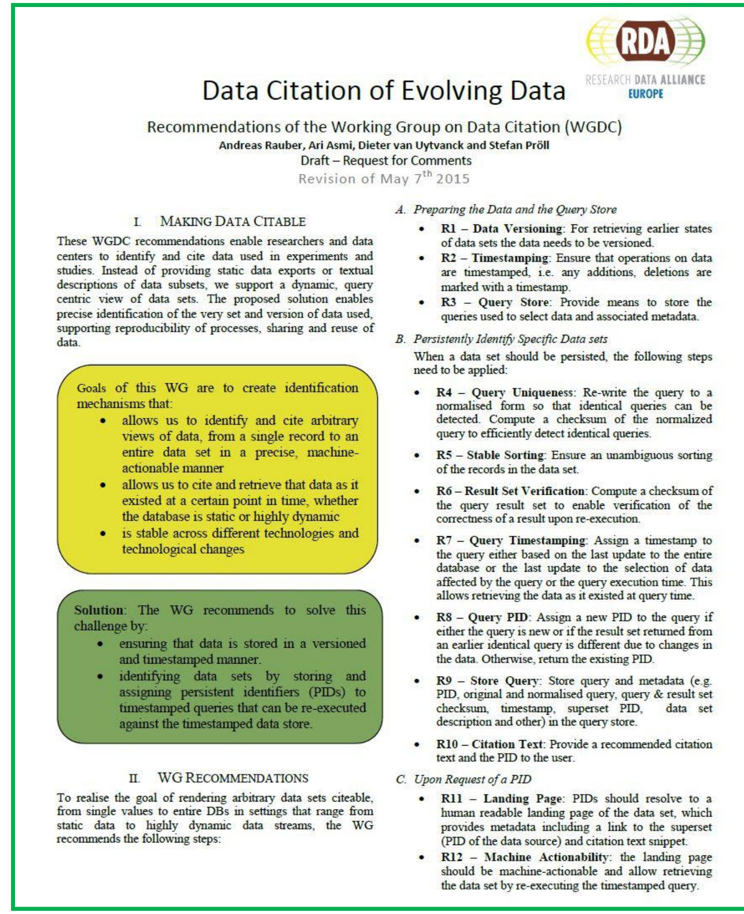
- ubset of data
- er gets
  - Data (pac
  - PID (e.g.
  - Hash valu
  - Recommended citati n text (e.g. bTeX)

This is an important advantage over traditional approaches relying on, e.g. storing a list of identifiers/DB dump!!!

- PID resolves to land g page
  - Provides detailed metadata, link parent data set, subset,…
  - Option to retrieve **original data** OR **current version** OR **changes**
- Upon activating PID associated with a data citation
  - Query is re-executed against time-stamped and versioned DB
  - Results as above are returned

Data Citation – Recommendations

- 2-page flyer,
  more extensive doc to follow
- **14 Recommendations**
- Grouped into **3 phases**:
  - Preparing data and query store
  - Persistently identifying specific data sets
  - Upon request of a PID
  - Upon modifications to the data infrastructure
- History
  - First presented March 30 2015
  - Major revision after workshop April 20/21
  - 2 upcoming webinars (June 9, June 24)



RESEARCH DATA ALLIANCE EUROPE

Data Citation of Evolving Data

Recommendations of the Working Group on Data Citation (WGDC)

Andreas Rauber, Ari Asmi, Dieter van Uytvanck and Stefan Pröll

Draft – Request for Comments

Revision of May 7th 2015

I. Making Data Citable

These WGDC recommendations enable researchers and data centers to identify and cite data used in experiments and studies. Instead of providing static data exports or textual descriptions of data subsets, we support a dynamic, query centric view of data sets. The proposed solution enables precise identification of the very set and version of data used, supporting reproducibility of processes, sharing and reuse of data.

Goals of this WG are to create identification mechanisms that:

- allows us to identify and cite arbitrary views of data, from a single record to an entire data set in a precise, machine-actionable manner
- allows us to cite and retrieve that data as it existed at a certain point in time, whether the database is static or highly dynamic
- is stable across different technologies and technological changes

Solution: The WG recommends to solve this challenge by:

- ensuring that data is stored in a versioned and timestamped manner.
- identifying data sets by storing and assigning persistent identifiers (PIDs) to timestamped queries that can be re-executed against the timestamped data store.

II. WG Recommendations

To realise the goal of rendering arbitrary data sets citeable, from single values to entire DBs in settings that range from static data to highly dynamic data streams, the WG recommends the following steps:

A. Preparing the Data and the Query Store

- **R1 – Data Versioning**: For retrieving earlier states of data sets the data needs to be versioned.
- **R2 – Timestamping**: Ensure that operations on data are timestamped, i.e. any additions, deletions are marked with a timestamp.
- **R3 – Query Store**: Provide means to store the queries used to select data and associated metadata.

B. Persistently Identify Specific Data sets

When a data set should be persisted, the following steps need to be applied:

- **R4 – Query Uniqueness**: Re-write the query to a normalised form so that identical queries can be detected. Compute a checksum of the normalized query to efficiently detect identical queries.
- **R5 – Stable Sorting**: Ensure an unambiguous sorting of the records in the data set.
- **R6 – Result Set Verification**: Compute a checksum of the query result set to enable verification of the correctness of a result upon re-execution.
- **R7 – Query Timestamping**: Assign a timestamp to the query either based on the last update to the entire database or the last update to the selection of data affected by the query or the query execution time. This allows retrieving the data as it existed at query time.
- **R8 – Query PID**: Assign a new PID to the query if either the query is new or if the result set returned from an earlier identical query is different due to changes in the data. Otherwise, return the existing PID.
- **R9 – Store Query**: Store query and metadata (e.g. PID, original and normalised query, query & result set checksum, timestamp, superset PID, data set description and other) in the query store.
- **R10 – Citation Text**: Provide a recommended citation text and the PID to the user.

C. Upon Request of a PID

- **R11 – Landing Page**: PIDs should resolve to a human readable landing page of the data set, which provides metadata including a link to the superset (PID of the data source) and citation text snippet.
- **R12 – Machine Actionability**: the landing page should be machine-actionable and allow retrieving the data set by re-executing the timestamped query.

Data Citat

Note:
- R1 & R2 are already pretty much standard in many (RDBMS-) research databases
- Different ways to implement
- A bit more challenging for some data types (XML, LOD, …)

A) Preparing the Data a

- **R1 – Data Versioning:** Apply versioning to ensure earlier states of data sets the data can be retrieved
- **R2 – Timestamping:** Ensure that operations on data are timestamped, i.e. any additions, deletions are marked with a timestamp
- **R3 – Query Store:** Provide means to store the queries used to select data and associated metadata

Data Citation – Recommendations

B) Persistently Identify Specific Data sets (1/2)

When a data set should be persisted:

- **R4 – Query Uniqueness:** Re-write the query to a normalised form so that identical queries can be detected. Compute a checksum of the normalized query to efficiently detect identical queries
- **R5 – Stable Sorting:** Ensure an unambiguous sorting of the records in the data set
- **R6 – Result Set Verification:** Compute a checksum of the query result set to enable verification of the correctness of a result upon re-execution
- **R7 – Query Timestamping:** Assign a timestamp to the query based on the last update to the entire database (or the last update to the selection of data affected by the query or the query execution time). This allows retrieving the data as it existed at query time

Data Citation – Recommendations

B) Persistently Identify Specific Data sets (2/2)

When a data set should be persisted:

- **R8 – Query PID:** Assign a new PID to the query if either the query is new or if the result set returned from an earlier identical query is different due to changes in the data. Otherwise, return the existing PID
- **R9 – Store Query:** Store query and metadata (e.g. PID, original and normalised query, query & result set checksum, timestamp, superset PID, data set description and other) in the query store
- **R10 – Citation Text:** Provide a recommended citation text and the PID to the user

Data Citation – Recommendations

C) Upon Request of a PID

- **R11 – Landing Page:** PIDs should resolve to a human readable landing page of the data set, which provides metadata including a link to the superset (PID of the data source) and citation text snippet

- **R12 – Machine Actionability:** the landing page should be machine-actionable and allow retrieving the data set by re-executing the timestamped query

Data Citation – Recommendations

D) Upon Modifications to the Data Infrastructure

- **R13 – Technology Migration:** When data is migrated to a new representation (e.g. new database system, a new schema or a completely different technology), the queries and associated checksums need to be migrated
- **R14 – Migration Verification:** Successful query migration should be verified by ensuring that queries can be re-executed correctly

Outline

- Joint Declaration of Data Citation Principles
- Challenges in non-trivial settings
- Recommendation of the RDA Working Group
- Benefits
- Summary

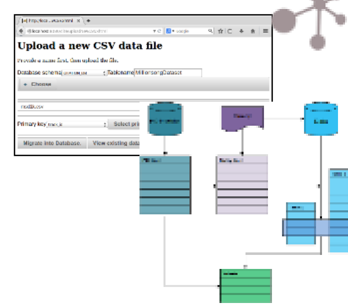
Benefits

- Retrieval of precise subset with minimal storage overhead
- Subset as cited or as it is now (including e.g. corrections)
- Query provides provenance information
- Checksums support verification
- Same principles applicable across all settings
  - Small and large data
  - Static and dynamic data
  - Different data representations (RDBMS, CSV, XML, LOD, …)
- Would work also for more sophisticated/general transformations on data

WG Pilots

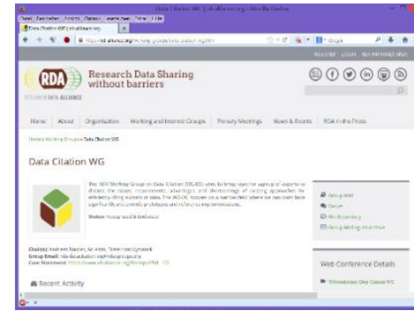
- Pilot workshops and implementations by
  - Various EU projects (TIMBUS, SCAPE,…)
  - NERC (UK Natural Environment Research Council Data Centres)
  - ESIP (Earth Science Information Partners)
  - CLARIN (XML, Field Linguistics Transcriptions)
  - Virtual Atomic and Molecular Data Centre

- Prototype solutions for
  - SQL, CSV, XML (partially)
  - LOD/RDF, triple-store DBs in the queue
  - Distributed data



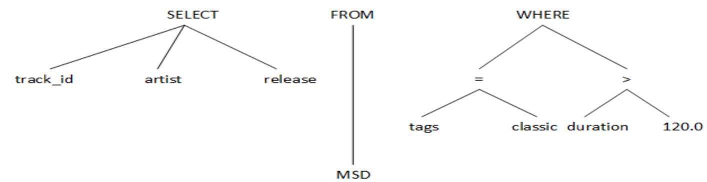
Join RDA and Working Group

If you are interested in joining the discussion, contributing a pilot, wish to establish a data citation solution, …



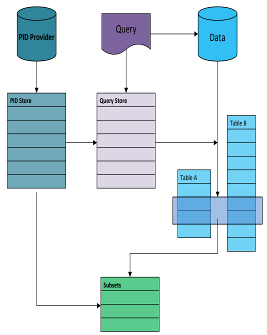
- Register for the RDA WG on Data Citation:
  - Website: https://rd-alliance.org/working-groups/data-citation-wg.html
  - Mailinglist: https://rd-alliance.org/node/141/archive-post-mailinglist
  - Web Conferences: https://rd-alliance.org/webconference-data-citation-wg.html
  - List of pilots: https://rd-alliance.org/groups/data-citation-wg/wiki/collaboration-environments.html

Thank you!

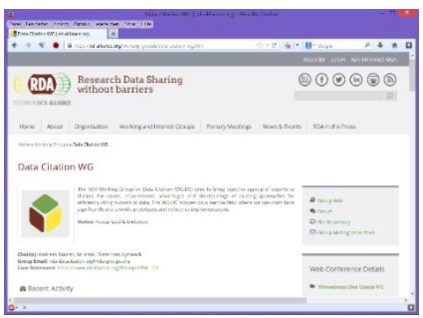


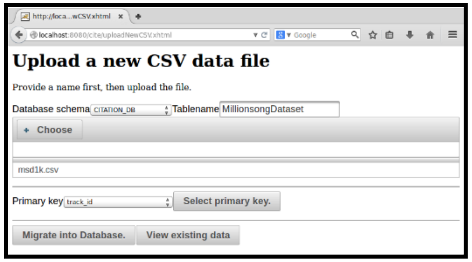

```
SELECT results.track_id, results.artist, results.release
        FROM MSD AS results JOIN (
                SELECT track_id, max(timestamp) AS latestTimestamp
                FROM MSD
                WHERE timestamp <= (SELECT @queryExecutionTimestamp)
                AND (track_id NOT IN
                        (SELECT track_id FROM MSD AS deletedRecords
                                WHERE deletedRecords.status_mark = 'deleted'
                                AND (deletedRecords.timestamp < @queryExecutionTimestamp))
                        )
        GROUP BY track_id
) AS version ON results.track_id = version.track_id AND results.timestamp = version.latestTimestamp

WHERE
    results.tags = 'classic'  AND results.duration> 120
ORDER BY results.track_id;
```



http://www.ifs.tuwien.ac.at/imp





Dynamic Data Citation for SQL Data

LNEC, MSD Implementation

SQL Prototype Implementation

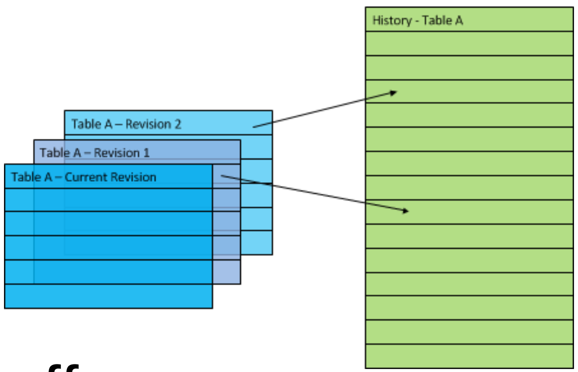
- LNEC Laboratory of Civil Engineering, Portugal
- Monitoring dams and bridges
- 31 manual sensor instruments
- 25 automatic sensor instruments
- Web portal
  - Select sensor data
  - Define timespans
- Report generation
  - Analysis processes
  - LaTeX
  - publish PDF report

SQL Prototype Implementation

- Million Song Dataset
  http://labrosa.ee.columbia.edu/millionsong/
- Largest benchmark collection in Music Retrieval
- Original set provided by Echonest
- No audio, only several sets of features
  (16 – 1440 measurements/features per song)
- Harvested, additional features and metadata extracted and offered by several groups
  e.g. http://www.ifs.tuwien.ac.at/mir/msd/download.html
- Dynamics because of metadata errors, extraction errors
- Research groups select subsets by genre, audio length, audio quality,…

SQL Time-Stamping and Versioning

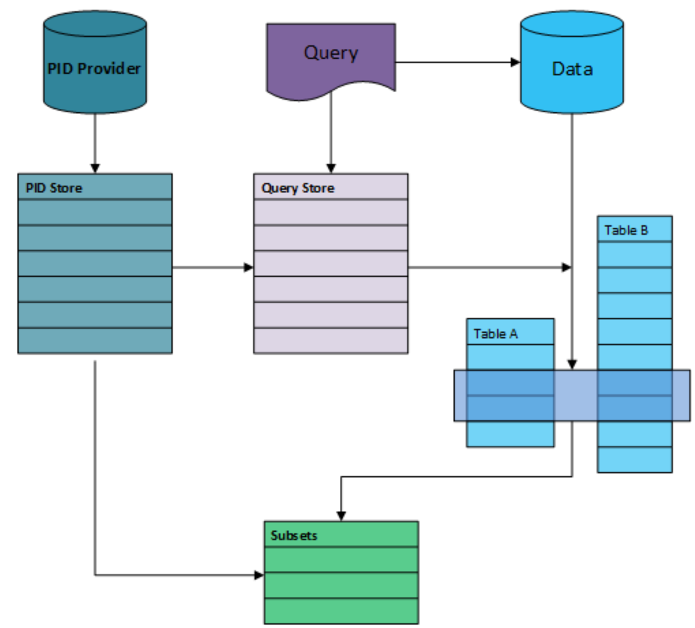
- Integrated
  - Extend original tables by temporal metadata
  - Expand primary key by record-version column
- Hybrid
  - Utilize history table for deleted record versions with metadata
  - Original table reflects latest version only
- Separated
  - Utilizes full history table
  - Also inserts reflected in history table



- Solution to be adopted depends on trade-off
  - Storage Demand
  - Query Complexity
  - Software adaption

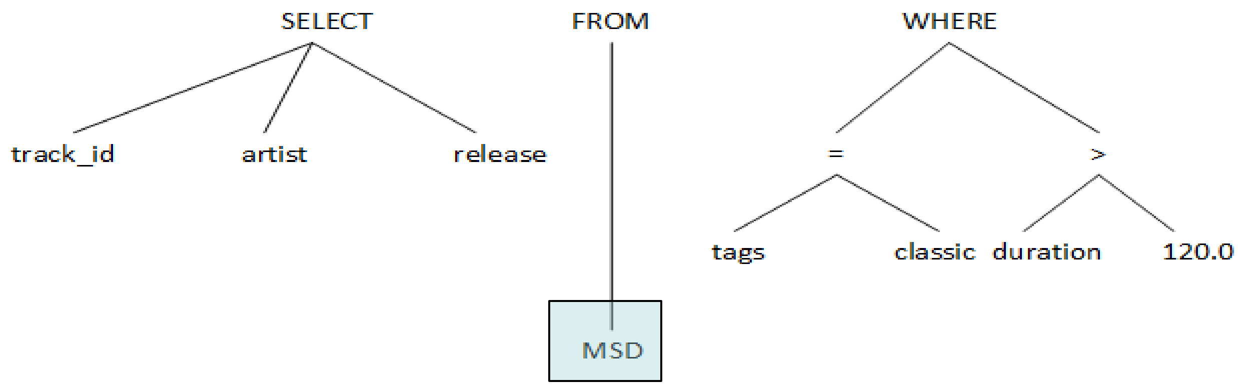
SQL: Storing Queries

- Add query store containing
  - PID of the query
  - Original query
  - Re-written query + query string hash
  - Timestamp
    (as used in re-written query)
  - Hash-key of query result
  - Metadata useful for citation / landing page
    (creator, institution, rights, …)
  - PID of parent dataset
    (or using fragment identifiers for query)



SQL Query Re-Writing

- Adapt query to history table




```sql
SELECT results.track_id, results.artist, results.release
        FROM MSD AS results JOIN (
                SELECT track_id, max(timestamp) AS latestTimestamp
                FROM MSD
                WHERE timestamp <= (SELECT @queryExecutionTimestamp)
                AND (track_id NOT IN
                        (SELECT track_id FROM MSD AS deletedRecords
                                WHERE deletedRecords.status_mark = 'deleted'
                                AND (deletedRecords.timestamp < @queryExecutionTimestamp))
                        )
        GROUP BY track_id
) AS version ON results.track_id = version.track_id AND results.timestamp = version.latestTimestamp

WHERE
    results.tags = 'classic'  AND results.duration> 120
ORDER BY results.track_id;
```

Dynamic Data Citation for CSV Data

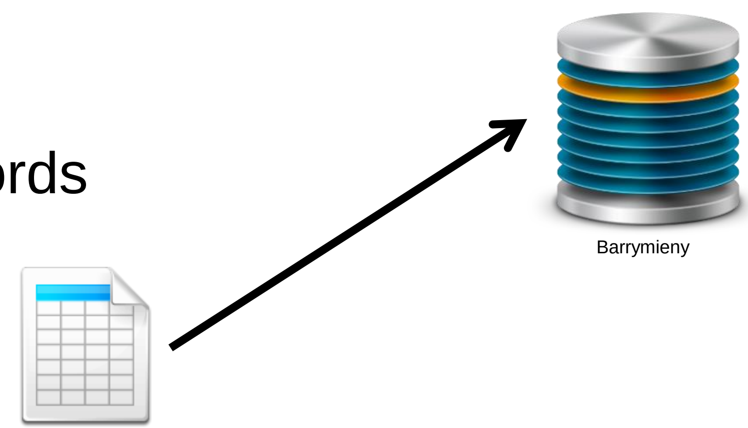
Open Source Reference Implementation

Dynamic Data Citation for CSV Data

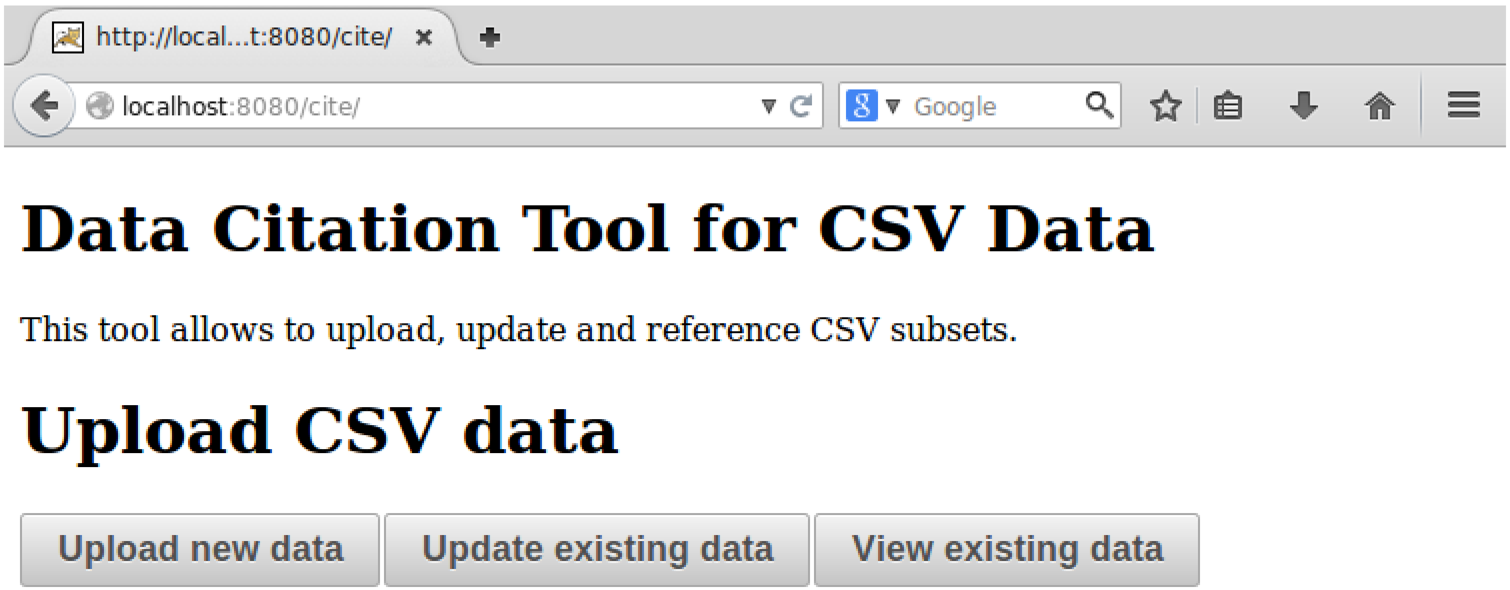
- Why CSV data? (not large, not very dynamic…)
  - Well understood and widely used
  - Simple and flexible
  - Most frequently requested during initial RDA meetings
- Goals:
  - Ensure cite-ability of CSV data
  - Enable subset citation
  - Support particularly small and large volume data
  - Support dynamically changing data
- 2 Options:
  - Versioning system (subversion/svn, git, …)
  - Migration to RDBMS

CSV Prototype: Basic Steps

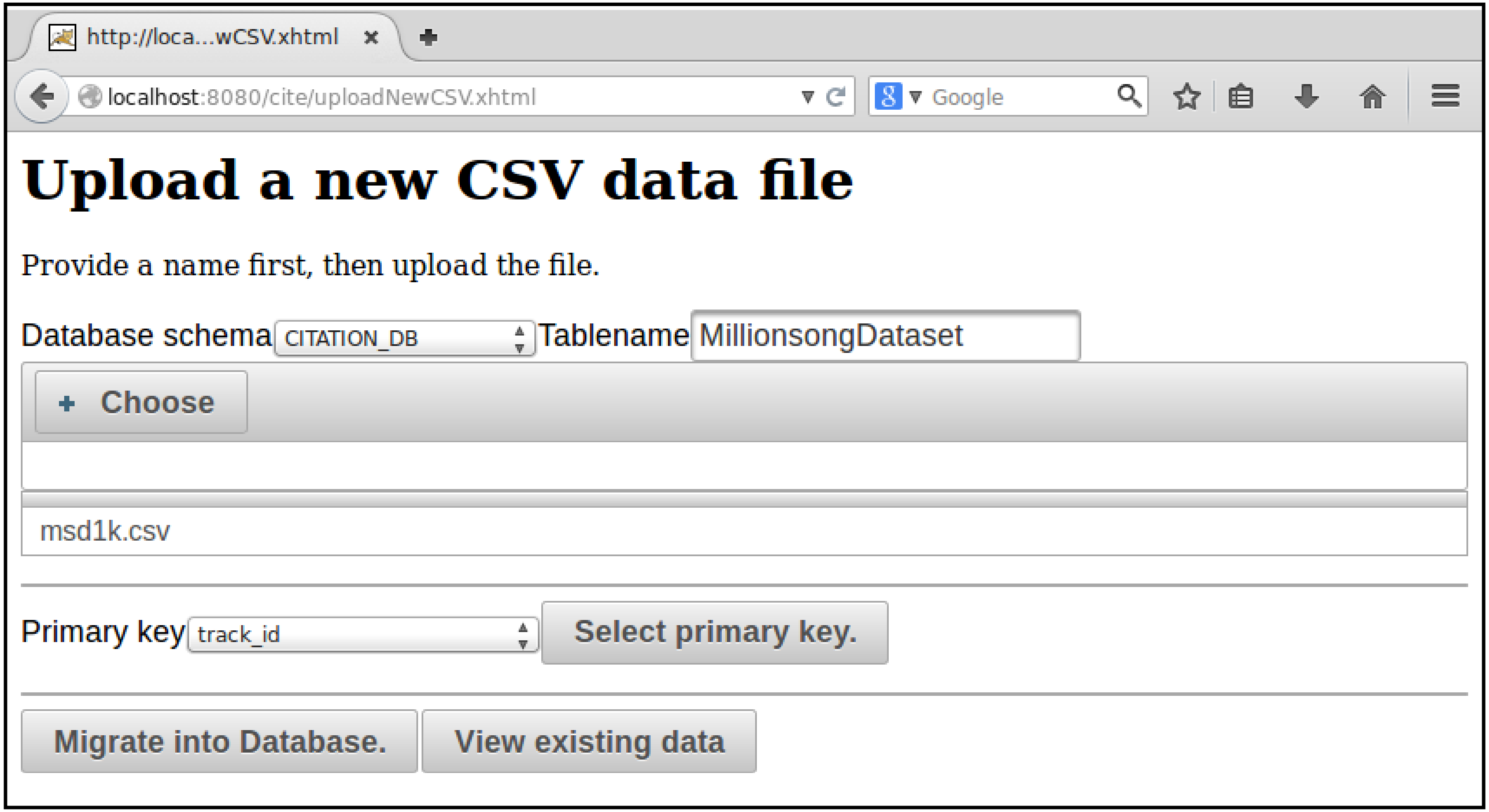
- Upload interface
  - Upload CSV files
- Migrate CSV file into RDBMS
  - Generate table structure, identify primary key
  - Add metadata columns for versioning
  - Add indices
- Dynamic data
  - Update / delete existing records
  - Append new data
- Access interface
  - Track subset creation
  - Store queries



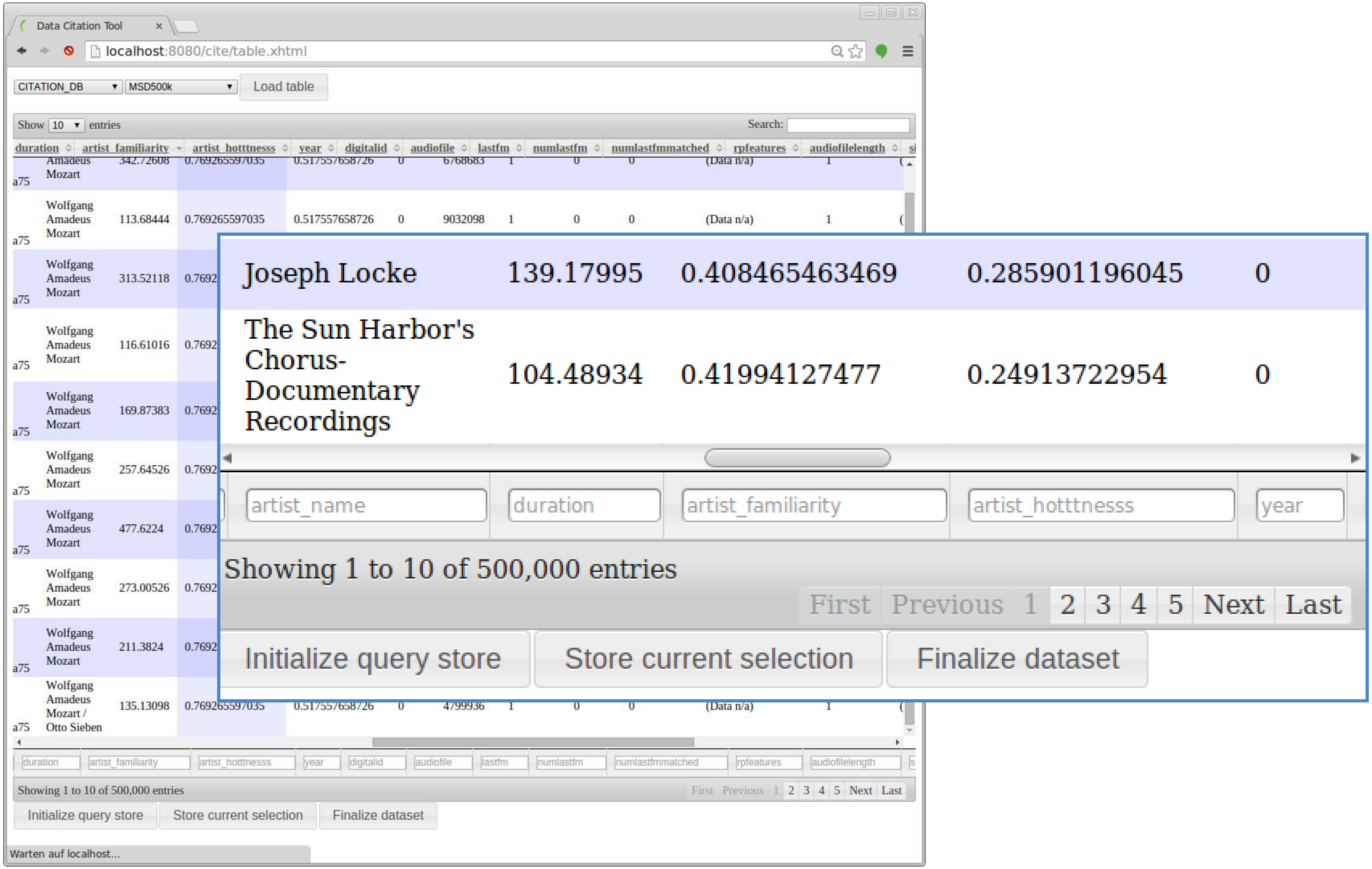
CSV Data Prototype



CSV Data Prototype



CSV Data Prototype



CSV Data Prototype

Suggested citation text:	Stefan Pröll (2015) "jj test" created at 2015-02-19 11:33:54.0, PID [ark:12345/5l86eH4qMX]. Subset of Stefan Pröll: "Adresses", PID [ark:12345/OjfL4gUmFo]

Download area

Download CSV Subset	Download	Download the CSV data of this subset at the execution time of the query
Download Latest Subset	Download	Download the CSV data of this subset at its current state
Download Full DB	Download	Download the full database as CSV file
Download Diff CSV file	Download	Download the differences as CSV between the subset at its original execution time and now.

CSV Data Prototype

SQL string

(innerSELECT.RECORD_STATUS = 'inserted' OR
innerSELECT.RECORD_STATUS = 'updated' AND
innerSELECT.LAST_UPDA
LAST_UPDATE) innerGrou
innerGroup.LAST_UPDATE
innerGroup.mostRecent W
UPPER('%jj%') ORDER B

Öffnen von _tmp_CSV-Files_CitationDB_stefan_ad... ×

Sie möchten folgende Datei öffnen:

...itationDB_stefan_adresses_12345-5l86eH4qMX.csv

Vom Typ: CSV-Dokument

Von: http://localhost:8080

Wie soll Firefox mit dieser Datei verfahren?

- (●) Öffnen mit LibreOffice Calc (Standard)
- ( ) DownThemAll!
- ( ) Datei speichern

[ ] Für Dateien dieses Typs immer diese Aktion ausführen

Abbrechen | OK

Suggested citation text:

Stefan Pröll (2015) "jj test" c ... eH4qMX].
Subset of Stefan Pröll: "Adr

Download area

Download CSV Subset	Download	Download the CSV data of this subset at the execution time of the query
Download Latest Subset	Download	Download the CSV data of this subset at its current state
Download Full DB	Download	Download the full database as CSV file
Download Diff CSV file	Download	Download the differences as CSV between the subset at its original execution time and now.

Progress update from VAMDC Distributed Data Centre

Carlo Maria Zwölf

Virtual Atomic and Molecular Data Centre

carlo-maria.zwolf@obspm.fr

VAMDC

- Virtual Atomic and Molecular Data Centre
- Worldwide e-infrastructure federating 41 heterogeneous and interoperable Atomic and Molecular databases
- Nodes decide independently about growing rate, ingest system, corrections to apply to already stored data
- Data-node may use different technology for storing data (SQL, No-sql, ASCII files),
- All implement VAMDC access/query protocols
- Return results in standardized XML format (XSAMS)
- Access directly node-by-node or via VAMDC portal, which relays the user request to each node

VAMDC

Workshop prior to RDA P4

Issues identified

- Each data node could modify/delete/add data without tracing
- No support for reproducibility of past data extraction

Proposed Data Citation WG Solution:

- Considering the distributed architecture of the federated VAMDC infrastructure, it seemed very complex to apply the “Query Store” strategy
  - Should we need a QS on each node?
  - Should we need an additional QS on the central portal?
  - Since the portal acts as a relay between the user and the existing nodes, how can we coordinate the generation of PID for queries in this distributed context?

VAMDC

Status / Progress since RDA P4

- Versioning adopted prior to P4
- Central service registering user interactions with data
- At each client SW notifies tracing service that a given **user** is using, at a given **time**, that specific **software** for submitting a given **query**
- Will assign single identifier for each unique query centrally
- Query store initially private (confidentiality issues)

Further Pilots

- NERC: UK Natural Environment Research Council
  - ARGO buoy network: SeaDataNet
  - Butterfly monitoring, Ocean buoy network, National hydrological archive, …
- ESIP: BCO-DMO
- XML Data in Field Linguistics (CLARIN, XBase)
- Further Pilots on XML, LOD, …
- Workshops:
  - NERC Workshop, London, July 1/2 2014
  - ESIP Mtg in Washington, Jan 8 2015: Earth Science Data
  - Data Citation Workshop, Riva di Garda, April 20/21
  - Bilateral meetings with data centers