Precise, Actionable Reference to Dynamic Data

Recommendations of the Working Group on Data Citation (WGDC)

Andreas Rauber, Mark Parsons, and the Working Group on Data Citation

Revision of original recommendation [1], April 4th 2025

I. PURPOSE

These recommendations enable researchers and data infrastructures to identify and cite the precise data used in experiments and studies. Instead of providing static data exports or textual descriptions of data subsets, the recommendations support a dynamic, query centric view of data sets. They enable unambiguous identification of the precise subset and version of data used, supporting reproducibility of processes using data and the sharing and reuse of data.

Goals of this WG are to create identification mechanisms that:

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

Solution: The WG recommends solving this challenge by:

- ensuring that data is stored in a versioned and timestamped manner.
- identifying data (sub)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 realize the goal of precisely identifying arbitrary subsets of data, from single values to entire data holdings in settings ranging from static data to highly dynamic data streams for any type of data, the WG recommends the adoption of the following actions:

A. Preparing the Data and the Query Store

Prepare existing data sources and provide the required infrastructure, which is needed for implementing the query-based approach as follows:

1. **Versioning and Timestamping:** IF data is evolving AND earlier states of the data should be reproducible, THEN the infrastructure MUST version changes to the data. It further MUST assign a timestamp of when a change becomes visible in the system.
2. **Query Store Facilities:** The infrastructure MUST provide means for storing queries and the associated metadata, specifically an execution timestamp, to allow re-execution against the then-valid state of the data source.

B. Persistently Identify Specific Data Sets

When a user selects (a subset of) data, this is referred to as a query executed against the data store. IF such a subset should be identifiable persistently, a data infrastructure SHOULD apply the following steps:

3. **Query Functionality Limitation:** A data infrastructure SHOULD limit the query to functions that can be deterministically re-computed across different technical infrastructure implementations.
4. **Query PID:** A data infrastructure MUST 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 of the identical query.
5. **Query Timestamping:** If the query received a new PID, the data infrastructure MUST assign a timestamp to the query based on the time the query was executed (or the last update to the data). This allows retrieving the data as it existed at the time a user issued a query.
6. **Consistent Sorting:** The data infrastructure SHOULD ensure that the sorting of the elements in any data set returned is unambiguous and reproducible. IF consistent sorting is not guaranteed the data infrastructure MUST be explicit about that.
7. **Result Set Verification:** The data infrastructure SHOULD compute fixity information (checksum) of the query result set to enable verification of the correctness of a result upon re- execution.
8. **Query Persistence:** The data infrastructure MUST persistently store the query and associated metadata (e.g. PID, original and normalized query, fixity information (if available), timestamp, database PID, data set description,

and other information) in the query store associated with the data queried and MUST assume responsibility for re-executing the query to reproduce the dataset when required to do so.

9. **Citation Texts:** The data infrastructure SHOULD generate citation texts in the format prevalent in the designated community to encourage scholarly citation of the data (e.g. Authors, Date, Dataset Title. PID).

C. *Resolving PIDs and Retrieving the Data*

10. **Landing Page:** PIDs MUST resolve to a human readable and machine-actionable landing page that provides a mechanism for accessing the data (via query re-execution) and metadata. The landing page MAY reveal the actual query being executed, as well as the associated timestamps and other metadata depending on client needs, security and privacy concerns.
11. **Data Access:** Data access MUST include the possibility to re-run the timestamped query against the accordant state of the data store. It SHOULD further provide the option to re-run the query against the current state of the data store and retrieve the semantically equivalent dataset but benefiting from all additions and corrections made (leading to a new PID being assigned if the result set differs). It MAY further offer the opportunity of retrieving a difference set of the changes that were made between the original execution timestamp and the current state, or re-executing the query against any arbitrary state (i.e. timestamp) of the data. The data infrastructure SHOULD verify the correctness of the re-execution using the fixity information stored with the original query execution and inform the client accordingly.

D. *Upon Modifications to the Data Infrastructure*

12. **Technology Migration:** When data is migrated to a new representation (e.g. new database system, a new schema or a completely different technology), the data infrastructure MUST migrate also the queries and associated fixity information and verify the correctness of the migration.

III. Benefits

The proposed solution has several benefits compared to current approaches relying on individual data exports for each data set or ambiguous natural language descriptions of data set characteristics.

- It allows identifying, retrieving and citing the precise data set with minimal storage overhead by only storing the versioned data and the queries used for creating the data set. In many environments data versioning is considered a best practice. Data subsets can be re- created on demand.
- It allows retrieving the data both as it existed at a given point in time as well as the current view on it, by re- executing the same query with the stored or current timestamp, thus benefiting from all corrections made since the query was originally issued. This allows tracing changes of data sets over the time and comparing the effects on the result set.
- The query stored as a basis for identifying the data set provides valuable provenance information on the way the specific data set was constructed, thus being semantically more explicit than a mere data export.
- The query store offers a valuable, central basis for analyzing data usage.
- Metadata such as checksums support the verification of correctness and authenticity of data sets retrieved.
- The recommendations are applicable across different types of data representation and data characteristics (big or small data; static or highly dynamic; identifying single values or the entire data set).
- If data is migrated to new representations, the queries can also be migrated, ensuring stability across changing technologies.
- Distributed data sources can rely on local timestamps at each node, avoiding the need for expensive synchronization in loosely coupled systems.

IV. Frequently Asked Questions

- May data be deleted? Yes, given appropriate policies. Queries may then not produce the same result set when re-executed. Landing pages should persist.
- Does the system need to store every query? No. Only data sets that should be persisted for citation / later re- use need to be stored. Persisting queries can be decided individually or policy-based in an automated fashion.
- Can I obtain only the most recent data set? Queries can, in principle, be re-executed with the original timestamp or with the current timestamp or any other timestamp desired. This allows retrieving the semantically identical data set but incorporating all changes, corrections or updates applied before the given timestamp.
- Which PID system should be used? Any PID system can be applied according to the institutional policy.
- How are the queries created? Queries can either be created manually via an interface/workbench or applications create the proper queries automatically. Any form of data access ultimately leads to a query, ranging from a single file access (retrieving the file's segments from a storage device and re-assembling them), to bounding boxes or slice/dice operations in multidimensional data cubes. All require the adaption of the query by adding metadata and timestamps.
- How can I share parts of my database? The query centric view allows selecting any particular view or data subset of the data from the complete data set, such as a time-delay embargo providing a view of the data exposing all changes made prior to a certain point in time.
- How does this support giving credit and attribution? Attribution and giving credit is supported via a provenance chain from a subset/view of data to the data set it was derived from and the data infrastructure hosting it, allowing to document intellectual contributions on the way. Analysis and recommendations on how to aggregate bibliometrics and credits is not addressed in the context of this WG.

V. References

[1] Rauber, Andreas, Ari Asmi, Dieter van Uytvanck, and Stefan Proell. "Data Citation of Evolving Data: Recommendations of the Working Group on Data Citation (WGDC)". Zenodo, October 20, 2015. https://doi.org/10.15497/RDA00016.

[2] Rauber, A., Göß wein, B., Zwölf, C. M., Schubert, C., Wörister, F., Duncan, J., Flicker, K., Zettsu, K., Meixner, K., McIntosh, L. D., Pröll, S., Miksa, T., & Parsons, M. A. (2021). Precisely and Persistently Identifying and Citing Arbitrary Subsets of Dynamic Data. Harvard Data Science Review, 3(4). https://doi.org/10.1162/99608f92.be565013