DrillBeyond: Open-World SQL Queries Using Web Tables

Julian Eberius, Maik Thiele, Katrin Braunschweig and Wolfgang Lehner

Database Technology Group
Department of Computer Science
Dresden University of Technology
D-01062 Dresden
{firstname.lastname}@tu-dresden.de

Abstract: The Web consists of a huge number of documents, but also large amounts structured information, for example in the form of HTML tables containing relational-style data. One typical usage scenario for this kind of data is their integration into a database or data warehouse in order to apply data analytics. However, in today's business intelligence tools there is an evident lack of support for so-called situational or ad-hoc data integration. In this demonstration we will therefore present *DrillBeyond*, a novel database and information retrieval engine which allows users to query a local database as well as the web datasets in a seamless and integrated way with standard SQL. The audience will be able to pose queries to our DrillBeyond system which will be answered partly from local data in the database and partly from datasets that originate from the Web of Data. We will demonstrate the integration of the web tables back into the DBMS in order to apply its analytical features.

1 Open-World SQL Queries

The system we want to demonstrate offers a novel way of integrating web tables into regular query processing in a relational database. We present a modified RDBMS that is able to answer so-called open-world queries which are not restricted to the schema of the local database. Instead the user is allowed to use arbitrary attribute names that do not appear in the original schema. Consider the following running example query:

```
SELECT population, n_name, AVG(o_totalprice)
    FROM nation
    JOIN region ON n_regionkey=r_regionkey
    JOIN customer ON n_nationkey=c_nationkey
    JOIN orders ON c_custkey=o_custkey
WHERE
    r_name = 'AMERICA'
GROUP BY population, n_name
ORDER BY population
```

The `population` attribute which is used in the `SELECT` and `ORDER BY` clauses is not part of the TPC-H schema and therefore requires special processing. In the DrillBeyond system, missing attributes are translated into keyword queries that are run against an index of open datasets on the web. It will answer the query by substituting the missing attribute

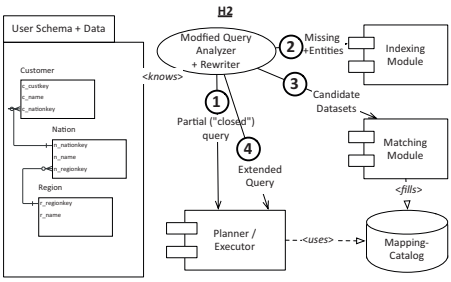

Figure 1: Architecture and Query Processing

Figure 2: Screenshot of the web front end used for the demonstration

values, e.g. for `population`, with values from the retrieved web datasets that are integrated into the local schema on the fly. In contrast to pure search systems, DrillBeyond is able to exploit the local schema and data as well as the context given by the SQL query to find the best matching web datasets. As usual with IR style approaches, the system will not be able to find the optimal dataset completely automatically, but instead needs to present the user with a ranked result list. Apart from identifying web datasets that can be potentially used to answer the open-world SQL query, the challenges lie in identifying the attributes of the web datasets that contain the correct values, as well as identifying those attributes that can be used to join the local table and the web data table.

2 System Architecture

The DrillBeyond system is implemented inside the open source RDBMS H2[1] by adding the following components: a *Modified Query Analyzer and Rewriter*, an *Indexing Module* and a *Schema/Instance Matching Module*. All other components of the system, such as the query optimizers, or join implementations, can be reused unmodified, as the output of the preprocessing steps is a standard SQL query referencing standard database objects. Figure 1 gives a global overview of the demo system, including the new DBMS components and query processing steps.

Modified Query Analyzer and Query Rewriting In a RDBMS, the query analyzer maps tokens from the SQL query to objects in the database, e.g., the token `n_name` to the corresponding attribute in the nation table (if this table is given in the respective `from` clause). If a token can not be mapped to a database object, an error is raised. For this demo, we modified H2's query analyzer to instead trigger a search for fitting datasets that can be used to answer the query. Specifically, the query processor will use the unknown token as

[1] `http://h2database.com`

one input for the search, but also the token's *context*, i.e., the related database objects and instance data. In our example, as the unknown `population` attribute is related to the `nation` table, we use instance data from the `nation` table to aid the search for a fitting join partner in the DrillBeyond index as described the following sections. Furthermore, the presented system also takes the specific query into account by applying local operations before looking for candidate datasets. In the running example, the selection on the region name is applied on the nation table before consulting the Indexing Module. In this way, the search for candidate datasets can return more specific results, while the amount of matching work that needs to be done can be minimized.
After collecting the set of relevant local tables and instances (the *context*), the analyzer calls the Matching and the Indexing Module, performs query rewriting and finally passes control to the regular optimizer and the executor. These new steps in query processing are depicted as numbers 1 to 3 in Figure 1. First, the extracted keyword token and its context are passed to the Indexing Module, which returns a list of candidate datasets. This list is passed to the Matching Module that checks which of those candidates can be joined with the local data at all, and if possible saves mappings in the catalog. In a third step the original query plan is rewritten to include a new attribute created from the candidates. In the following sections, we will give more details on the indexing and matching modules as well as on the web front used in the demo.

Indexing Module The Index Module keeps an local index of web datasets. It indexes the datasets metadata, such as title, context and attribute names as well as the text column values. For this demo, the index is realized using Lucene which supports, among others features, normalization/stemming and boolean keyword queries. We added synonym expansion via WordNet[2] to be able to identify additional candidate datasets. A lookup in the index is performed using the unknown tokens and their context, as passed from the query analyzer. The result ranking is a mixture of classical keyword-search ranking, e.g., comparing the query tokens to dataset metadata and attribute names, but also instance-based techniques, which are applied in the Matching Module to refine the ranking. Continuing our running example, the index lookup will identify the queried term `population`, its synonyms, and the selected American nation names such as Argentina, Brazil or the US, in several datasets, and pass them to the next stage.

Matching Module Since there are no foreign-key relationships between the local and open datasets, a join can only be performed when at least one matching column pair of the local table and the respective open dataset can be identified. Therefore, the *Matching Module* employs a set of classic schema and instance matching techniques, such as string similarity measures and external knowledge such as synonym dictionaries. By doing this we are able to rank the result candidates produced by the index lookup and to prune all datasets which can not be joined. The ranking is influenced mainly by the quality of the mapping, e.g., the monogamy and coverage of the created mapping between two datasets. If a join candidate is found, the instance level mappings between the matching attributes are stored in the mapping catalog to establish a foreign key relationship between the two

[2]http://wordnet.princeton.edu/

datasets. The stored mappings are later used to perform the joins to produce the actual query result. Continuing the example, the Matching module will bridge differences between the country names in TPCH and the candidate datasets, and will also prune datasets that do not contain all the necessary countries, e.g., datasets only about South American countries, as the mapping coverage will be lower in these cases.

Web Frontend In addition to the H2 back end, we implemented a web front end which enables the interactions with the user, such as presentation of the search results and selection of a fitting dataset from the candidates. Figure 2 gives an impression of this interface. The users can enter regular SQL queries and browse different variations of the query results, depending on which candidate dataset is used to answer the query. For each candidate, the front end will display the dataset's schema, the attributes matching the local table, the attributes (potentially) containing the missing values as well as sample rows of the query result when the respective candidate is chosen. Finally, for each open dataset the available metadata as indexed by the Indexing module can be viewed. This allows the users to make a more informed choice about which open datasets to use to complement their data.

3 Demo Walkthrough

In this live demonstration, users will be able to get a feeling for the potential of Web Data in data analytics by posing SQL queries including undefined attributes to the DrillBeyond system. Using an console or the web interface, they will be able to choose from different preloaded local databases to perform analysis on. The preloaded schemata include TPC-H and an IMDb sample. Then, the users can enter SQL queries on the chosen schemata, using open attributes as they see fit. The demo system will consult its index of open datasets, which for this demo, contains about one million web tables extracted from the English version of Wikipedia. Depending on the tool used, the audience will be able to study query results as one raw SQL result table on the console, or as individual query results depending on a selected candidate when using the web interface as shown in Figure 2. In the web front end they also have access to the metadata, schema and sample rows of the candidate datasets. A screencast demonstrating both the raw SQL console as well as the web front end is available on the web[3].
Please note that this demonstration is an extended version of [ETBL12] presented at VLDB'12.

References

[ETBL12] Julian Eberius, Maik Thiele, Katrin Braunschweig, and Wolfgang Lehner. DrillBeyond: Enabling Business Analysts to Explore the Web of Open Data. *PVLDB*, 5(12):1978–1981, 2012.

[3] `http://wwwdb.inf.tu-dresden.de/edyra/DrillBeyond`