Data Posting: a New Frontier for Data Exchange in the Big Data Era*

Domenico Saccà and Edoardo Serra

DIMES, Università della Calabria, 87036 Rende, Italy
sacca@unical.it, eserra@deis.unical.it

1 Preliminaries on Data Exchange and Count Constraints

Data exchange [5, 1] is the problem of migrating a data instance from a source schema to a target schema such that the materialized data on the target schema satisfies a number of given integrity constraints (mainly inclusion and functional dependencies). The target schema typically contains some new attributes that are defined using existentially quantified variables: the main issue is to reduce arbitrariness in selecting such variable values. Therefore a data exchange solution is required to be "universal" in the sense that homomorphisms exists into every possible solution, i.e., a universal solution enjoys a sort of "minimal arbitrariness" property. The main research goal of the large data exchange literature is to single out situations for which a universal solution exists and can be computed in polynomial time.

Recently a different approach to data exchange has been proposed in [11] that considers a new type of data dependency, called *count constraint* (an extension of cardinality constraint), that prescribes the result of a given count operation on a relation to be within a certain range. We illustrate this approach by means of an example. Consider a source relation scheme S with the following attributes: I (Item), B (Brand), P (Price). The target scheme is the relation scheme T with attributes: I, B, P, W (Warehouse), C (product Category), R (price Range). We assume that the domains of I, B and P for T are the projections of the source relation S on the respective attributes, e.g., $\mathcal{D}_{\mathtt{B}} = \pi_{\mathtt{B}}(\mathtt{S})$. Also the domains of the other attributes of T are finite and are defined by supplementary source relations: $\mathcal{D}_{\mathtt{W}}$ (the list of all available warehouses for storing items), $\mathcal{D}_{\mathtt{C}}$ (a 2-arity relation listing the category for each product) and $\mathcal{D}_{\mathtt{R}}$ (a 3-arity relation listing all price ranges together with the interval extremes). The mapping is defined as follows (as usual, lower-case and upper-case letters denote variables that are respectively universally and existentially quantified – in addition, dotted letters denote free variables used for defining sets):

(1): $S(\mathtt{i},\mathtt{p},\mathtt{b}) \wedge \mathcal{D}_{\mathtt{C}}(\mathtt{i},\mathtt{c}) \wedge \mathcal{D}_{\mathtt{R}}(\mathtt{r},\underline{\mathtt{p}},\overline{\mathtt{p}}) \wedge (\underline{\mathtt{p}} \leq \mathtt{p} < \overline{\mathtt{p}}) \rightarrow \mathtt{1} \leq \#(\{\ddot{\mathtt{W}} : T(\mathtt{i},\mathtt{p},\mathtt{b},\ddot{\mathtt{W}},\mathtt{c},\mathtt{r})\}) \leq \mathtt{5}$

(2): $T(\_,\_,\mathtt{b},\mathtt{w},\_,\_) \rightarrow \#(\{\ddot{\mathtt{I}} : T(\ddot{\mathtt{I}},\mathtt{P},\mathtt{b},\mathtt{w},\mathtt{C},\mathtt{R})\}) \geq \mathtt{5}$

(3): $T(\_,\_,\_,\mathtt{w},\mathtt{c},\_) \rightarrow \#(\{\ddot{\mathtt{I}} : T(\ddot{\mathtt{I}},\mathtt{P},\mathtt{B},\mathtt{w},\mathtt{c},\mathtt{R})\}) \geq \mathtt{5}$

(Note that # is an interpreted function symbol for computing the cardinality of a set, existentially quantified variables are local in a set term, i.e., $\{\ddot{\mathtt{I}} : \mathtt{T}(\ddot{\mathtt{I}},\mathtt{P},\mathtt{b},\mathtt{w},\mathtt{C},\mathtt{R})\}$

* The research was partially funded by MIUR (PON Project "InMoto – Information Mobility for Tourism") and by Calabria Region (POR Regional Innovation Pole on ICT).

stands for $\{\ddot{\mathtt{I}} : \exists\, \mathtt{P}\, \mathtt{C}\, \mathtt{R}\, \mathtt{T}(\ddot{\mathtt{I}}, \mathtt{P}, \mathtt{b}, \mathtt{w}, \mathtt{C}, \mathtt{R})\}$, and that $T(\_, \_, \mathtt{b}, \mathtt{w}, \_, \_)$ and $T(\_, \_, \_, \mathtt{w}, \mathtt{c}, \_)$ stand for the projections of T on $\mathtt{B}, \mathtt{W}$ and on $\mathtt{W}, \mathtt{C}$ respectively.) The values of the new attributes C and R are univocally determined by the domains $\mathcal{D}_{\mathtt{C}}$ and $\mathcal{D}_{\mathtt{R}}$, whereas the values for the attribute W may be arbitrarily taken from the domain $\mathcal{D}_{\mathtt{W}}$ provided that the following count constraints are satisfied: (1) every item must be stored into at least one and at most 5 warehouses, (2)-(3) if a warehouse stores an item of a given brand (resp., category) , it must store at least other 4 items of the same brand (resp., category).

The approach of [11] has received three main strong criticisms from a number of reviewers of the paper during its long process for publication: (1) the high complexity of deciding whether an admissible solution for T exists (NEXP-complete under combined complexity), (2) the lack of a universal solution and (3) an alleged fictitious nature of data exchange with count constraints.

The first criticism is a common drawback for many approaches – e.g., data exchange is undecidable in the general case. The real (and open) issue is: are there tractable (and, at the same time, meaningful) cases? Even more, as we shall argue later in the paper, we believe that intractability must be dealt with more and more every day, as it is currently done in many knowledge discovery tasks.

The second criticism is motivated by the relevance of a universal solution to answer certain queries. Nevertheless, at the risk to be accused of heresy, we believe that answering certain queries is not a "must" for data exchange. In fact, one may want to get answers from the transferred data without caring about certainty w.r.t. source data. Indeed, as discussed in [10], within a scenario of privacy-preserving data exchange, one could even have an opposite goal: defining a target schema for which answering a number of given "sensible" queries is "uncertain"! As the main point of our data exchange setting is the choice of some attribute values, we have presented in [10] an extension of Datalog to provide an alternative formalization of the problem. To give an intuition, the above constraint (1) can be expressed using Datalog with choice as follows:

$$\mathtt{T}(\mathtt{I}, \mathtt{P}, \mathtt{B}, \mathtt{W}, \mathtt{C}, \mathtt{R}) \leftarrow \mathtt{S}(\mathtt{I}, \mathtt{P}, \mathtt{B}), \mathcal{D}_{\mathtt{C}}(\mathtt{I}, \mathtt{C}), \mathcal{D}_{\mathtt{R}}(\mathtt{r}, \underline{\mathtt{P}}, \overline{\mathtt{P}}), \underline{\mathtt{P}} \leq \mathtt{P} < \overline{\mathtt{P}}, \mathcal{D}_{\mathtt{W}}(\mathtt{W}), \mathtt{choice}(\mathtt{I}, \mathtt{W})[5]$$

The construct $\mathtt{choice}(\mathtt{I}, \mathtt{W})[5]$ extends the classical choice construct: instead of choosing exactly one value of $\mathtt{W}$ for each $\mathtt{I}$, the new construct enables the selection of up to five distinct values.

Concerning the criticism about the fictitious nature of data exchange with count constraints, indeed the original formulation of our approach intended to address the problem of generating fact tables (i.e., relations used in OLAP applications) satisfying a number of given count constraints, mainly to perform benchmark experiments on artificial data cubes reflecting patterns extracted from reality. On the sidelines of our work, we eventually realized that the same setting can be used for a new declination of data exchange to transform database relations into Web contents.

2 Data Posting: a New Paradigm for Sharing Data in Big Data Platforms

It is well known that a Web Search Engine such as Google mainly executes string (word) selection queries across public resources on the Web. In a sense, for those who have

spent decades of their research effort to elaborate query languages advancing SQL, it may be frustrating to eventually witness the victory of query languages that are much more elementary than SQL, as they only enable to list words for making a simple selection. The complexity is behind the query language: the large technological infrastructure for crawling, indexing and accessing huge amount of contents on the Web. We have added "may be" to the above-mentioned frustration as we believe that advanced query languages may take their revenge if moved to the backstage: they may have a crucial role in enriching the semantics of database tuples that are posted on the Web cloud. We fully agree with the arguments of [3] and [2]: to solve the challenges of the emerging "Big Data" platforms, database technology (including database theory) may continue to have a crucial role, if it will be suitably revised and immersed into the new technological and applicative perspectives. For instance, much effort is being presently put in providing intelligent answers to simple queries, see the numerous semantic Web proposals and, among them, the *Knowledge Graph* of Google [12]. The database community has significant expertise in declarative data processing and how to do it efficiently as well as to make it scale. The community should apply its expertise to the design of more intelligent and more efficient future Big Data platforms.

Since recently, Google provides an interesting solution, the *Google Search Appliance* (GSA), to access public and private contents, such as emails and database tuples, that cannot be directly browsed by the search engine. To this end, so-called *connectors* extend the reach of the GSA to non-Web repositories [6]. For instance database tuples can be materialized as XML documents by ad-hoc connectors and, afterwards, transformed into HTML documents for the search. A database connector can be thought of as a simple exchange data setting for posting data on the Web.

Inspired by this solution, we have shaped our vision on data posting: during the process of database publishing, the contents can be enriched by supplying additional concepts. In our example, we added values derived from a sort of "ontology" for the classification of products and of prices. We have also added the attribute `W` (warehouse) together with its domain, but in this case the values for the attribute are not predetermined by the available domain values but are selected on the basis of the constraints. We point out that the issue of inventing new values to be included into the target relation is one of the goals of classical data exchange setting. The main difference with data posting is the focus: to preserve the relationships with the source database, classical data exchange only considers dependencies delivering universal solutions, whereas we look for more expressive constraints for enriching the contents of the exchanged data at the cost of loosing certainty. As witnessed by data mining applications, the process of knowledge discovery is inherently uncertain.

Another important peculiarity of data posting is the structure of the target database scheme. We assume that it consists of a unique relation scheme that corresponds to a *flat fact table* as defined in *OLAP analysis* – we recall that an OLAP system is characterized by multidimensional data cubes that enable manipulation and analysis of data stored in a source database from multiple perspectives (see for instance [4]).

A *fact table* is relation scheme whose attributes are *dimensions* (i.e., properties, possibly structured at various levels of abstraction) and *measures* (i.e., numeric values) – but measures can be seen as dimensions as well. For instance, in our example, the

attribute P (price) is a typical measure but together with R (price range) it forms a pair of 2-layered dimensions. In general, in addition to a fact table, an OLAP scheme includes other tables describing dimension attributes (e.g., star or snowflake scheme [4]). We instead denormalize all tables into a unique flat fact table in order to comply with search engine strategies – string selection queries are easier to express on denormalized tables and can be massively parallelized on them (perhaps, after almost thirty years, the time has come for revenge of the Universal Relation [9]). We observe that the very objective of Google Knowledge Graph is to add hidden dimensions to a fact table (corresponding to a Web document) on-the-fly during the search in order to enrich the semantics of strings[1].

Let $\mathbf{S} = \langle S_1, \ldots, S_n \rangle$ be a *source database schema* with relation schemes $S_1, \ldots, S_n$, $\boldsymbol{\mathcal{D}} = \langle \mathcal{D}_1, \ldots, \mathcal{D}_m \rangle$ be a *domain database schema* with domain relation schemes $\mathcal{D}_1, \ldots, \mathcal{D}_m$ and T be a *target flat fact table* whose attribute domains are in $\boldsymbol{\mathcal{D}}$.

A *source-to-target count constraint* is a dependency over $\langle \mathbf{S}, \boldsymbol{\mathcal{D}}, T \rangle$ of the form $\forall \mathbf{x} \, (\, \phi(\mathbf{x}) \rightarrow \psi_T(\mathbf{x}) \,)$, where $\mathbf{x}$ is a list of variables, the formula ϕ is a conjunction of atoms with predicate symbols in $\mathbf{S} \cup \boldsymbol{\mathcal{D}}$ and whose variables are exactly the ones in $\mathbf{x}$, and ψ_T is a formula of the form:

$$\beta_1 \leq \#(\{\mathbf{y} : \exists \mathbf{z} \, \alpha(\mathbf{x}, \mathbf{y}, \mathbf{z})\}) \leq \beta_2.$$

The above formula is a 3-arity comparison predicate, where β_1 and β_2 are simple terms that are either constants or variables in $\mathbf{x}$, the two lists $\mathbf{y}$ and $\mathbf{z}$ consist of distinct variables that are also different from the ones in $\mathbf{x}$, the formula $\alpha(\mathbf{x}, \mathbf{y}, \mathbf{z})$ is a conjunction of atoms $T(\mathbf{x}, \mathbf{y}, \mathbf{z})$ and $\#$ is an interpreted function symbol that computes the cardinality of the (possibly empty) set defined by $\{\mathbf{y} : \exists \mathbf{z} \, \alpha(\mathbf{x}, \mathbf{y}, \mathbf{z})\}$.

A *target count constraint* differs from a source-to-target constraint only in the formula ϕ, which in this case is is a conjunction of atoms with T as predicate symbol.

Given finite source instances I_S for $\mathbf{S}$, $I_{\mathcal{D}}$ for $\boldsymbol{\mathcal{D}}$ and I_T for T, $\langle I_S, I_{\mathcal{D}}, I_T \rangle$ satisfies a count constraint if for each substitution $\mathbf{x}/\mathbf{v}$ that makes true $\phi(\mathbf{x})[\mathbf{x}/\mathbf{v}]$, the cardinality of the set W is in the range between $\beta_1[\mathbf{x}/\mathbf{v}]$ and $\beta_2[\mathbf{x}/\mathbf{v}]$, where W is the (possibly empty) projection on $\mathbf{y}$ of the selection of I_T defined by $\{\mathbf{y} : \exists \mathbf{z} \, \alpha(\mathbf{x}, \mathbf{y}, \mathbf{z})\}[\mathbf{x}/\mathbf{v}]$.

Observe that a *Tuple Generating Dependency* (TGD) $\forall \mathbf{x} \, (\, \phi(\mathbf{x}) \rightarrow \exists \mathbf{y} \, \psi_T(\mathbf{x}, \mathbf{y}) \,)$ of the classical data exchange setting can be formulated by the following count constraint:

$$\forall \mathbf{x} \, (\, \phi(\mathbf{x}) \rightarrow \#(\, \{\mathbf{y} : \psi_T(\mathbf{x}, \mathbf{y})\} \,) \geq 1 \,).$$

Also an *Equality Generating Dependency* (EGD) $\forall \mathbf{x} (\, \psi_T(\mathbf{x}) \rightarrow x_1 = x_2 \,)$, where x_1 and x_2 are variables in $\mathbf{x}$, can be formulated by the following count constraint:

$$\forall \mathbf{x} \, (\, true \rightarrow \#(\{y : \phi(\mathbf{x}) \wedge (y = x_1 \vee y = x_2)\} \,) \leq 1)$$

where y is a new variable not included in $\mathbf{x}$. The extension of our formalism to include "safe" comparison predicates such as $(y = x_1 \vee y = x_2)$ is straightforward.

We are now ready to formulate the data posting problem:

[1] Actually we were tempted to title this paper: "The Elegant Search Universe: Superstrings, Hidden Dimensions and the Quest for the Ultimate Big Data Theory", but we later discovered that a similar title was already used by a physics book. Ugh, we are always late!

Definition 1. The *data posting setting* $(\mathbf{S}, \boldsymbol{\mathcal{D}}, T, \Sigma_{st}, \Sigma_t)$ consists of a source database schema S, a domain database scheme $\boldsymbol{\mathcal{D}}$, a target flat fact table T, a set Σ_{st} of source-to-target count constraints and a set Σ_t of target count constraints. The *data posting problem* associated with this setting is: given finite source instances I_S for $\mathbf{S}$ and $I_{\mathcal{D}}$ for $\boldsymbol{\mathcal{D}}$, find a finite instance I_T for T such that $\langle I_S, I_{\mathcal{D}}, I_T \rangle$ satisfies both Σ_{st} and Σ_t. □

The main difference w.r.t. classical data exchange is the presence of the domain database scheme that stores "new" values (dimensions) to be added while exchanging data. In a motto we can say that "data posting is enriching data while exchanging them".

Theorem 1. *Given* $(\mathbf{S}, \boldsymbol{\mathcal{D}}, T, \Sigma_{st}, \Sigma_t)$ *and finite source instances* I_S *for* $\mathbf{S}$ *and* $I_{\mathcal{D}}$ *for* $\boldsymbol{\mathcal{D}}$*, the problem of deciding whether there exists an instance* I_T *of* T *such that* $\langle I_S, I_{\mathcal{D}}, I_T \rangle$ *satisfies* $\Sigma_{st} \cup \Sigma_t$ *is* NEXP*-complete under the combined complexity and* NP*-complete under the data complexity.* □

The proof of NEXP-completeness can be easily derived from a similar result presented in [11] – actually NEXP-completeness also holds for binary domains. The proof of NP-completeness consists of a rather straightforward reduction from the graph 3-coloring problem. We stress that our complexity results derives from the assumption that the domains of the attributes in T are finite and are part of the input.

3 Research Lines for Data Posting

In our example we have used count constraints to perform an elementary task of grouping items into warehouses on the basis of their categories and brands. Grouping objects is the goal of two important data mining techniques: clustering and classification [7]. A first research line is to include some features of these techniques in the data posting setting. This is coherent with our ambitious goal of posting data with knowledge value added. And this explains why we do not insist in finding tractable cases: most of the data mining problems are indeed intractable but yet there are a lot of approaches aimed at finding solutions for small-sized or well-structured instances or at searching for approximate solutions.

In [11] we have shown that a version of count constraint implementing the "group-by" operator can be used to mimic another classical data mining technique: frequent itemset mining. As an example, the following "group-by" count constraint imposes the items a, b and c to be together frequent in the warehouses (the frequency threshold is fixed at 3):

$$\mathtt{x} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\} \ \rightarrow \#(\{\ddot{\mathtt{W}} : \mathtt{x} \subseteq \{\ddot{\mathtt{I}} : \ T(\ddot{\mathtt{I}}, \mathtt{P}, \mathtt{B}, \ddot{\mathtt{W}}, \mathtt{C}, \mathtt{R})\}\}) \geq 3$$

In a "group-by" count constraint, the formula $\#(\{\mathbf{y} : \exists \mathbf{z}\, \alpha(\mathbf{x}, \mathbf{y}, \mathbf{z})\})$ is replaced by $\#(\{\mathbf{y} : t * \{\, \mathbf{z_1} : \exists \mathbf{z_2}\ \alpha(\mathbf{x}, \mathbf{y}, \mathbf{z})\, \}\, \})$, where $*$ is a set operator such as $=$ or $\subset$ or $\subseteq$, and t is a bounded set term.

Continuing in our efforts to add semantics to data posting, we point out that in our example we have used hierarchy domains C and R to classify items. We dared to say with a bit of shame that the two domains represent a sort of ontology. A second line of research is to add "real" ontology tools to the data posting setting.

Let us now discuss the issue of relaxing the assumption that the domains of all attributes of the target relation T are finite and are explicitly listed in the input. Consider first the case that one of such domains, say D, is still finite but their values are not listed. Thus only the cardinality k of the domain is given in input; in addition, the k values can be generated using a polynomial-time function. This case has been considered in [11] to show that, in the presence of "group-by" count constraints, also data complexity of decision data posting becomes NEXP-complete. A third research line is to analyze the undecidabity risk of the case that D is a countably infinite set – obviously a polynomial-time function for generating domain values must be available.

We conclude by pointing out some possible interesting relationships between data posting and data integration. It is known that a target instance need not be materialized in data integration; the main focus there is on answering queries posed over the target schema using views that express the relationship between the target and source schemata [8]. Data posting can be thought of as a bottom-up enriched view directly provided by information source experts, which should be integrated with the classical top-down local view designed by the data integration administrator.

References

1. ARENAS, M., BARCELÓ, P., FAGIN, R., AND LIBKIN, L. Locally consistent transformations and query answering in data exchange. In *PODS* (2004), C. Beeri and A. Deutsch, Eds., ACM, pp. 229–240.
2. BORKAR, V. R., CAREY, M. J., AND LI, C. Inside "Big Data Management": Ogres, Onions, or Parfaits? In *EDBT* (2012), E. A. Rundensteiner, V. Markl, I. Manolescu, S. Amer-Yahia, F. Naumann, and I. Ari, Eds., ACM, pp. 3–14.
3. CATTELL, R. Scalable SQL and NoSQL data stores. *SIGMOD Record 39*, 4 (2010), 12–27.
4. CHAUDHURI, S., AND DAYAL, U. An Overview of Data Warehousing and OLAP Technology. *SIGMOD Record 26*, 1 (1997), 65–74.
5. FAGIN, R., KOLAITIS, P. G., AND POPA, L. Data Exchange: getting to the core. *ACM Trans. Database Syst. 30*, 1 (2005), 174–210.
6. GOOGLE DOCUMENTATION. Getting the Most from Your Google Search Appliance. In *Google Developers Site* (November, 2011). https://developers.google.com/search-appliance/documentation/614/QuickStart/quick_start_intro.
7. JIAWEI HAN, MICHELINE KAMBER, J. P. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, 2011.
8. LENZERINI, M. Data Integration: A Theoretical Perspective. In *PODS* (2002), L. Popa, S. Abiteboul, and P. G. Kolaitis, Eds., ACM, pp. 233–246.
9. MAIER, D., ULLMAN, J. D., AND VARDI, M. Y. On the Foundations of the Universal Relation Model. *ACM Trans. Database Syst. 9*, 2 (1984), 283–308.
10. SACCÀ, D., AND SERRA, E. Data Exchange in Datalog Is Mainly a Matter of Choice. In *Datalog* (2012), P. Barceló and R. Pichler, Eds., vol. 7494 of *Lecture Notes in Computer Science*, Springer, pp. 153–164.
11. SACCÀ, D., SERRA, E., AND GUZZO, A. Count Constraints and the Inverse OLAP Problem: Definition, Complexity and a Step toward Aggregate Data Exchange. In *FoIKS* (2012), T. Lukasiewicz and A. Sali, Eds., vol. 7153 of *Lecture Notes in Computer Science*, Springer, pp. 352–369.
12. SINGHAL, A. Introducing the Knowledge Graph: things, not strings. In *Official Google Blog* (May, 2012). http://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html.