# On the Role of Integrity Constraints in Data Integration

Andrea Calì, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini
Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113, I-00198 Roma, Italy
{lastname}@dis.uniroma1.it

**Abstract**

*We discuss the issue of dealing with integrity constraints over the global schema in data integration. On the one hand, integrity constraints can be used to extract more information from incomplete sources, similarly to the case of databases with incomplete information. On the other hand, integrity constraints raise the problem of dealing with the inconsistency of the whole system, due to contradictory data at the sources. We also present a data integration system developed by taking into account such issues.*

## 1 Introduction

Integrating heterogeneous data sources is a fundamental problem in databases, which has been studied extensively in the last two decades both from a formal and from a practical point of view, cf. [1]. Recently, also driven by the need to integrate data sources on the Web, much of the research on integration has focussed on so called *data integration* [2, 1]. Data integration is the problem of combining the data residing at different sources, and providing the user with a unified view of these data, called global (or mediated) schema, over which queries to the data integration system are expressed. A data integration system has therefore to free the user from the knowledge on which sources contain the data of interest, how such data are structured at the sources, and how such data are to be merged and reconciled to answer her/his queries. A crucial issue in data integration is how elements of the global schema and element at the sources are mapped to each other. In particular, two basic approaches have been proposed: *global-as-view* (GAV) and *local-as-view* (LAV) [2, 1]. The first one requires that the elements of the global schema are expressed as a view (a query) over the elements at the sources [3, 4, 5]. The second one requires that the elements at the sources are expressed as a view over the elements in the global schema [6, 7, 8]. Commonly, it is assumed that the GAV approach leads to simpler query answering mechanisms, while the LAV approach makes it easier to add or remove sources from the system. On the other hand, the LAV approach requires more sophisticated query answering techniques that are related to query answering in the presence of incomplete information [7, 9, 10, 1]. This distinction however becomes blurred as soon as we introduce integrity constraints in the global schema (see later).

A fundamental assumption related to mapping global schema and sources to each other is whether the data at the sources are considered *exact* or just *sound* in the mapping. That is, do the views defining the mapping capture exactly the relation between the data at the sources and those in the global schema (i.e., is the mapping


**Bulletin of the IEEE Computer Society Technical Committee on Data Engineering**

exact), or are the data provided by the sources only part of the data that should populate a given element of the global schema (i.e., is the mapping sound)?

Since the global schema acts as the interface to the user for query formulation, it should incorporate flexible and powerful representation mechanisms to relate the various elements of the global schema according to the semantics of the domain of interest. *Integrity constraints* play exactly this role by asserting the interrelations among the elements in a schema. The importance of allowing for integrity constraints in the global schema has been stressed in several works on data integration [1, 8, 11].

Introducing integrity constraints on the global schema has a deep impact on the data integration system and the query answering mechanisms. Indeed, the data retrieved from the sources may or may not fulfill the constraints in the global schema. Moreover, if the retrieved data do not fully satisfy the constraints, in principle, it may still be possible to complete the retrieved data (provided the mapping is sound) so as to satisfy the constraints. Or, vice-versa, it may not be possible to do so. In this case one may still be interested in the part of the data that does not violate the constraints. In other words, introducing integrity constraints raises issues related to query answering in the presence of incomplete information [12] (just as in the LAV approach) and to query answering in the presence of inconsistent information [13, 14, 15].

As a result of this observation, we have a spectrum of possible data integration systems, which can be classified according to the direction of the mapping (LAV/GAV), the type of the mapping (exact/sound), and the presence and type of integrity constraints allowed in the global schema. In this paper we concentrate on the GAV approach only, and we survey the general issues raised in the various cases of the above classification. To make the discussion concrete we also present an implemented data integration system based on the relational model, that follows the GAV approach, assumes sources to be sound, and allows for key and foreign key constraints in the global schema. Notably, the system takes into account such constraints in the query answering process.

## 2 Framework for Data Integration

We illustrate a framework for data integration that accounts for the presence of integrity constraints in the global schema. We formalize a *data integration system* $\mathcal{I}$ in terms of a triple $\langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$, where

- $\mathcal{G}$ is the *global schema*, which provides a reconciled, integrated, and virtual view of the underlying sources, in terms of which users query the integration system. We assume in the discussion that such a global schema is expressed as a relational schema, possibly including integrity constraints, such as key constraints, foreign key constraints, and general inclusion dependencies.

- $\mathcal{S}$ is the *source schema*, which describes the structure of the set of sources accessible by the integration system. We assume that the sources are wrapped so as to be viewed as relations.

- $\mathcal{M}$ is the *mapping* between $\mathcal{G}$ and $\mathcal{S}$, which establishes the connection between the elements of the global schema and those of the sources. We restrict our attention to mappings of type GAV, which associate to each relation $g$ in $\mathcal{G}$ a query $\mathcal{M}(g)$ over $\mathcal{S}$. For each element $g$ of the global schema, a further specification (which may be either $exact$ or $sound$, see below) is associated to $\mathcal{M}(g)$ [7, 9, 10].

Given the data at the sources, we specify which data satisfy the global schema, compatibly with the data at the sources and the mapping. Let us denote by $q^{\mathcal{DB}}$ the answer set of a query $q$ over a database $\mathcal{DB}$. A *source database* $\mathcal{D}$ for a data integration system $\mathcal{I} = \langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$ is constituted by one relation $r^{\mathcal{D}}$ for each source $r$ in $\mathcal{S}$. We call *(global) database for* $\mathcal{I}$ any database $\mathcal{B}$ for $\mathcal{G}$, and such a database is said to be *legal* for $\mathcal{I}$ wrt to $\mathcal{D}$ if:

- $\mathcal{B}$ satisfies the integrity constraints of $\mathcal{G}$, and
- $\mathcal{B}$ satisfies $\mathcal{M}$ with respect to $\mathcal{D}$, i.e., for each relation $g$ in $\mathcal{G}$ we have that: if $\mathcal{M}(g)$ is *exact*, then $\mathcal{M}(g)^{\mathcal{D}} = g^{\mathcal{B}}$; if $\mathcal{M}(g)$ is *sound*, then $\mathcal{M}(g)^{\mathcal{D}} \subseteq g^{\mathcal{B}}$.

Note that in general there is more than one database that is legal for $\mathcal{I}$ wrt $\mathcal{D}$, hence the semantics must be specified in terms of a set of databases rather than a single one. The consequences of this will be discussed in the next section.

Queries to a data integration system $\mathcal{I} = \langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$ are posed in terms of the relations in $\mathcal{G}$, and are intended to provide the specification of which data to extract from the virtual database represented by $\mathcal{I}$. The task of specifying which tuples are in the answer to a query is complicated by the existence of several legal global databases, and this requires to introduce the notion of certain answers. A tuple $t$ is a *certain answer* to a query $q$ wrt a source database $\mathcal{D}$, if $t \in q^{\mathcal{B}}$ for *all* global databases $\mathcal{B}$ that are legal for $\mathcal{I}$ wrt $\mathcal{D}$.

**Example 1:** Let $\mathcal{I}^1 = \langle \mathcal{G}^1, \mathcal{S}^1, \mathcal{M}^1 \rangle$ be a data integration system where $\mathcal{G}^1$ has the relation schemas $\mathsf{student}(Scode, Sname, Scity)$, $\mathsf{university}(Ucode, Uname)$ and $\mathsf{enrolled}(Scode, Ucode)$, and the constraints

$$\begin{array}{rcl} key(\mathsf{student}) & = & \{Scode\} \\ key(\mathsf{university}) & = & \{Ucode\} \\ key(\mathsf{enrolled}) & = & \{Scode, Ucode\} \end{array} \qquad \begin{array}{rcl} \mathsf{enrolled}[Scode] & \subseteq & \mathsf{student}[Scode] \\ \mathsf{enrolled}[Ucode] & \subseteq & \mathsf{university}[Ucode] \end{array}$$

$\mathcal{S}^1$ consists of three sources. Source $\mathsf{stu}$, of arity 4, contains information about students with their code, name, city, and date of birth. Source $\mathsf{univ}$, of arity 2, contains codes and names of universities. Finally, Source $\mathsf{enr}$, of arity 2, contains information about enrollment of students in universities. The mapping $\mathcal{M}^1$ is defined by

$$\begin{array}{rl} \mathcal{M}^1(\mathsf{student}): & \mathsf{student}(x,y,z) \leftarrow \mathsf{stu}(x,y,z,w) \\ \mathcal{M}^1(\mathsf{university}): & \mathsf{university}(x,y) \leftarrow \mathsf{univ}(x,y) \\ \mathcal{M}^1(\mathsf{enrolled}): & \mathsf{enrolled}(x,w) \leftarrow \mathsf{enr}(x,w) \end{array}$$

# 3 The Role of Integrity Constraints in Query Answering

The ultimate goal of a data integration system is to answer queries posed by the user in terms of the global schema. Since data are stored at the sources, query answering requires to reformulate the query in terms of the source schema, taking into account the mapping. It follows that the nature of the mapping has a great influence on the whole process. We now discuss this issue, by pointing out the impact of integrity constraints in the reformulation step. We focus on the GAV approach only. We distinguish among four cases, depending on whether the mapping is sound or exact, and whether the global schema contains integrity constraints or not.

The first case we consider is the one with *exact mapping*, and with *no integrity constraints* in the global schema. The exact mapping provides a unique way to determine which are the data of the global database on the basis of the data at the sources. Since such data cannot contradict the global schema, in this case, the semantics of the data integration system is characterized by a single global database, namely the one that is obtained by associating to each element $g$ the set of tuples computed by the query $\mathcal{M}(g)$ over the sources. It follows that query answering can be based on a simple *unfolding* strategy: when processing a query $q$ over the global schema, every element in $q$ is substituted with the corresponding query over the sources, and the resulting query is evaluated at the sources. Most GAV data integration systems, such as TSIMMIS [3], Garlic [16], Squirrel [17], and MOMIS [18], follows this approach.

The second case is the one with *exact mapping*, and *with integrity constraints* in the global schema. Again, the exact mapping precisely specifies which are the data of the global database in terms of the data at the sources. If such data are coherent with respect to the integrity constraints of the global schema, query answering can be carried out as if there were no constraints. However, differently from the previous case, it might be that the data retrieved from the sources do not satisfy the integrity constraints in the global schema. This situation is similar to the case where we aim at answering queries over a single inconsistent database. This problem is studied in several papers, including [13, 14, 15, 19]. Obviously, all the above mentioned papers start from the observation that the usual first-order semantics is not suited for this case, and tackle the fundamental problem of devising a meaningful semantics for query answering. In general, the role of such a semantics is to single out which are the reasonable “repairs” of the global database. Intuitively, a *repair* of a database $\mathcal{B}$ is a global database that satisfies the integrity constraints of the global schema, and can be obtained from $\mathcal{B}$ by a minimum number of insertions and deletions [20]. One of the basic problems is that, depending on the types of integrity constraints, there might be more than one repair. It follows that a simple unfolding strategy is likely to be inappropriate for the new semantics, and, indeed, recent papers aim at defining new query answering methods that are sound and complete with respect to the new formal framework [19].

The third case is the one with *sound mapping*, and with *no integrity constraints*. In this case, the mapping does not determine a unique global database on the basis of the source data, and indeed, we must accept all global databases that are supersets of the data retrieved from the sources. Formally, this is similar to the case where we have a database with incomplete information (or, a partially specified database). In principle, query answering can be affected by such an incompleteness. However, when the queries posed to the global schema are monotone (in particular, when they do not contain negation), in computing certain answers one can exploit the fact that there is a unique “minimal” global database, which is the one obtained by associating to each element of the global schema the set of tuples computed by the corresponding query. In other words, a simple unfolding strategy is again sufficient for this case.

The last case is the one with *sound mapping*, and *with integrity constraints* in the global schema. In this case, given the nature of the mapping, the semantics of the data integration system is specified in terms of a set of global databases, namely, those databases that are supersets of the data retrieved from the sources. However, among all the elements of this set, we must select only those databases that satisfy the integrity constraints in the global schema. If there are no such global databases, we basically fall into the second case, in the sense that we have to consider the notion of repair to make query answering meaningful. If there is at least one global database that is coherent with the mapping, and satisfies the integrity constraints of the global schema, then the query processing technique should be able to select only those answers that are true in all such global databases. Obviously, the characteristic of query processing will strongly depend upon the types of integrity constraints specified in the global schema. In the next section, a query processing strategy that is able to deal with the class of foreign key constraints is shown. Notice that in such a case the unfolding strategy proves to be too naïve, as shown by the following example.

**Example 2:** Referring to Example 1, where key and foreign key constraints are present in the global schema, suppose to have the following source database $\mathcal{D}$:

$\mathsf{stu}^{\mathcal{D}}$ :

| | | | |
|---|---|---|---|
| 12 | *anne* | *florence* | 21 |
| 15 | *bill* | *oslo* | 24 |

$\mathsf{univ}^{\mathcal{D}}$ :

| | |
|---|---|
| $AF$ | *bocconi* |
| $BN$ | *ucla* |

$\mathsf{enr}^{\mathcal{D}}$ :

| | |
|---|---|
| 12 | $AF$ |
| 16 | $BN$ |

Due to the integrity constraints in $\mathcal{G}^1$, 16 is the code of some student. Observe, however, that nothing is said by $\mathcal{D}$ about the name and the city of such a student. Therefore, we must accept as legal all databases that differ in such attributes of the student with code 16. Note that this is a consequence of the assumption of having sound views. If we had exact or complete views, this situation would have lead to an inconsistency of the data integration system. Instead, when dealing with sound views, we can think of extending the data contained in the sources in order to satisfy the integrity constraint over the global schema. The fact that, in general, there are several possible ways to carry out such an extension implies that there are several legal databases for the data integration systems. Now, consider the query

$$\mathsf{q}(x) \leftarrow \mathsf{student}(x, y, z) \wedge \mathsf{enrolled}(x, w)$$

The correct answer to the query is $\{12, 16\}$, because, due to the integrity constraints in $\mathcal{G}^1$, we know that 16 appears in the first attribute of $\mathsf{student}$ in all the databases for $\mathcal{I}^1$ that are legal wrt $\mathcal{D}$. However, we do not get this information from $\mathsf{stu}^{\mathcal{D}}$, and, therefore, a simple unfolding strategy retrieves only the answer $\{12\}$ from $\mathcal{D}$, thus proving insufficient for query answering in this case. Notice that, if the query asked for the student name instead of the student code (i.e., the head is $\mathsf{q}(y)$ instead of $\mathsf{q}(x)$), then one could *not* make use of the dependencies to infer additional answers.

## 4 The IBIS Data Integration System

To make the discussion concrete, we now present how the above concepts have been realized in a data integration system called IBIS [21, 22], which was developed by CM Sistemi in collaboration with the University of Rome "La Sapienza". IBIS is based on the relational model, and allows for key and foreign key constraints in the global schema. IBIS adopts the GAV approach for defining the mapping between the global schema relations and the sources. It deals with heterogeneous data sources, such as relational databases, legacy data sources, and web pages. Non-relational data sources are suitably wrapped so as to present themselves to the query precessing subsystem as relational sources (possibly with binding pattern, but we will ignore this aspect in the following).

The GAV mapping between the relations in the global schema and the tables at the sources is realized through arbitrary queries. This generality allows us to incorporate in such queries data cleaning mechanisms [23] to resolve conflicts on keys. Indeed it has been shown that, if key conflicts are resolved by the extraction process from the sources, then the other constraints in the global schema, namely the foreign key constraints, although tuple generating, will not cause further violation of key constraints. Observe that, by adopting this solution, IBIS is essentially delegating to the designer of the GAV mapping the responsibility of dealing with key conflicts.

The IBIS query processing algorithm fully takes into account foreign key constrains, and this allows the system to return answers to queries that depend on joins on attribute values that are not stored in the sources, but whose existence is guaranteed by the foreign key constraints (cf. Example 2). Differently from key constraints, the system deals with foreign key constraints in a completely automated way. It is possible to show that the IBIS query processing algorithm produces exactly the set of certain answers to queries, i.e., it is sound and complete with respect to the semantics of the data integration system. The IBIS query processing algorithm is constituted by three conceptually separate phases (in fact, the system is highly optimized and does not necessarily keep such phases distinct in performing its computations):

1. the query is *expanded* to take into account foreign key constraints in the global schema;
2. the atoms in the expanded query are *unfolded* according to their definition in terms of the mapping, obtaining a query expressed over the sources;
3. the expanded and unfolded query is *executed* over the sources, to produce the answer to the original query.

Unfolding and execution are standard steps in GAV query processing (although in IBIS they are more involved than usual because they perform data cleaning to resolve conflicts on keys). The expansion phase is the distinguishing feature of query processing in IBIS. The expansion is performed by viewing each foreign key constraint $r_1[\vec{x}] \subseteq r_2[\vec{y}]$, where $\vec{x}$ and $\vec{y}$ are sets of $h$ attributes and $\vec{y}$ is a key for $r_2$, as a logic programming rule

$$r_2'(\vec{x}, f_{h+1}(\vec{x}), \ldots, f_n(\vec{x})) \leftarrow r_1'(\vec{x}, x_{h+1}, \ldots, x_m)$$

where each $f_i$ is a Skolem function, $\vec{x}$ is a vector of $h$ variables, and we have assumed for simplicity that the attributes involved in the foreign key are the first $h$ ones. Each $r_i'$ is a predicate, corresponding to the global relation $r_i$, defined by the above rules for foreign key constraints, together with the rule

$$r_i'(x_1, \ldots, x_n) \leftarrow r_i(x_1, \ldots, x_n)$$

Once such a logic program $\Pi_{\mathcal{G}}$ has been defined, it can be used to generate the expanded query associated to the original query $q$. This is done by performing a partial evaluation [24] wrt $\Pi_{\mathcal{G}}$ of the body of $q'$, which is the query obtained by substituting in $q$ each predicate $r_i$ with $r_i'$. In the partial evaluation tree, a node is not expanded anymore either when: (i) no atom in the node unifies with a head of a rule; or (ii) when the Skolem functions have appeared in the substitutions for the distinguish variables, which makes the current answer unsuitable for being returned; or (iii) when the node is subsumed by (i.e., is more specific than) one of its predecessors, since the node cannot provide any answer that is not already provided by its more general predecessor [21]. Variants of the IBIS query processing mechanism have also been studied for other kinds of integrity constraints [25].

View-based query answering is known to be tightly related to query containment [7, 26]. Query containment in the relational setting in presence of inclusion dependencies alone, and in presence of key-based constraints and inclusion dependencies of a special form has been studied in [27], using chase-based techniques. However, more work has to be done to understand whether such techniques can be adapted for doing query answering in data integration systems, since a straightforward use of such techniques would require to materialize the whole global database, which may not feasible in practice.

Finally, as mentioned, the presence of integrity constraints in the global schema blurs the distinctions between GAV and LAV. In particular, [28] shows that a LAV system based on the relational model where the LAV mapping is defined in terms of conjunctive queries over the global schema, can be rephrased into a (query) equivalent GAV system that includes, in the global schema, inclusion dependencies together with a simple class of equality-generating dependencies.

## 5 Conclusions

In this paper we have investigated how in data integration the presence of integrity constraints on the global schema has a significant impact on the semantics of a data integration system. The presence of such constraints, even of a simple form, raises the need of dealing with incomplete information and possibly with inconsistencies. These issues confirm that data integration is a rich and challenging research topic [1].

## References


[1] Lenzerini, M.: Data integration: A theoretical perspective. In: Proc. of PODS 2002. (2002) 233–246

[2] Halevy, A.Y.: Answering queries using views: A survey. VLDB Journal **10** (2001) 270–294

[3] Garcia-Molina, H., Papakonstantinou, Y., Quass, D., Rajaraman, A., Sagiv, Y., Ullman, J.D., Vassalos, V., Widom, J.: The TSIMMIS approach to mediation: Data models and languages. J. of Intelligent Information Systems **8** (1997) 117–132

[4] Tomasic, A., Raschid, L., Valduriez, P.: Scaling access to heterogeneous data sources with DISCO. IEEE Trans. on Knowledge and Data Engineering **10** (1998) 808–823

[5] Goh, C.H., Bressan, S., Madnick, S.E., Siegel, M.D.: Context interchange: New features and formalisms for the intelligent integration of information. ACM Trans. on Information Systems **17** (1999) 270–293

[6] Kirk, T., Levy, A.Y., Sagiv, Y., Srivastava, D.: The Information Manifold. In: Proceedings of the AAAI 1995 Spring Symp. on Information Gathering from Heterogeneous, Distributed Enviroments. (1995) 85–91

[7] Abiteboul, S., Duschka, O.: Complexity of answering queries using materialized views. In: Proc. of PODS'98. (1998) 254–265

[8] Calvanese, D., De Giacomo, G., Lenzerini, M., Nardi, D., Rosati, R.: Data integration in data warehousing. Int. J. of Cooperative Information Systems **10** (2001) 237–271

[9] Grahne, G., Mendelzon, A.O.: Tableau techniques for querying information sources through global schemas. In: Proc. of ICDT'99. Volume 1540 of LNCS., Springer (1999) 332–347

[10] Calvanese, D., De Giacomo, G., Lenzerini, M., Vardi, M.Y.: Query processing using views for regular path queries with inverse. In: Proc. of PODS 2000. (2000) 58–66

[11] Fernandez, M.F., Florescu, D., Levy, A., Suciu, D.: Verifying integrity constraints on web-sites. In: Proc. of IJCAI'99. (1999) 614–619

[12] van der Meyden, R.: Logical approaches to incomplete information. In Chomicki, J., Saake, G., eds.: Logics for Databases and Information Systems. Kluwer Academic Publisher (1998) 307–356

[13] Lin, J., Mendelzon, A.O.: Merging databases under constraints. Int. J. of Cooperative Information Systems **7** (1998) 55–76

[14] Arenas, M., Bertossi, L.E., Chomicki, J.: Consistent query answers in inconsistent databases. In: Proc. of PODS'99. (1999) 68–79

[15] Greco, G., Greco, S., Zumpano, E.: A logic programming approach to the integration, repairing and querying of inconsistent databases. In: Proc. of ICLP'01. Volume 2237 of LNAI., Springer (2001) 348–364

[16] Tork Roth, M., Arya, M., Haas, L.M., Carey, M.J., Cody, W.F., Fagin, R., Schwarz, P.M., II, J.T., Wimmers, E.L.: The Garlic project. In: Proc. of ACM SIGMOD. (1996) 557

[17] Zhou, G., Hull, R., King, R., Franchitti, J.C.: Using object matching and materialization to integrate heterogeneous databases. In: Proc. of CoopIS'95. (1995) 4–18

[18] Beneventano, D., Bergamaschi, S., Castano, S., Corni, A., Guidetti, R., Malvezzi, G., Melchiori, M., Vincini, M.: Information integration: the MOMIS project demonstration. In: Proc. of VLDB 2000. (2000)

[19] Lembo, D., Lenzerini, M., Rosati, R.: Source inconsistency and incompleteness in data integration. In: Proc. of KRDB 2002. (2002)

[20] Fagin, R., Ullman, J.D., Vardi, M.Y.: On the semantics of updates in databases. In: Proc. of PODS'83. (1983) 352–365

[21] Calì, A., Calvanese, D., De Giacomo, G., Lenzerini, M.: Data integration under integrity constraints. In: Proc. of CAiSE 2002. Volume 2348 of LNCS., Springer (2002) 262–279

[22] Calì, A., Calvanese, D., De Giacomo, G., Lenzerini, M., Naggar, P., Vernacotola, F.: IBIS: Data integration at work. In: Proc. of SEBD 2002. (2002) 291–298

[23] Bouzeghoub, M., Lenzerini, M.: Introduction to the special issue on data extraction, cleaning, and reconciliation. Information Systems **26** (2001) 535–536

[24] De Giacomo, G.: Intensional query answering by partial evaluation. J. of Intelligent Information Systems **7** (1996) 205–233

[25] Calì, A., Calvanese, D., De Giacomo, G., Lenzerini, M.: Accessing data integration systems through conceptual schemas. In: Proc. of ER 2001. (2001) 270–284

[26] Calvanese, D., De Giacomo, G., Lenzerini, M., Vardi, M.Y.: View-based query answering and query containment over semistructured data. In: Proc. of DBPL 2001. (2001)

[27] Johnson, D.S., Klug, A.C.: Testing containment of conjunctive queries under functional and inclusion dependencies. J. of Computer and System Sciences **28** (1984) 167–189

[28] Calì, A., Calvanese, D., De Giacomo, G., Lenzerini, M.: On the expressive power of data integration systems. In: Proc. of ER 2002. (2002)