Empirical evidence for the usefulness of Armstrong relations in the acquisition of meaningful functional dependencies

Warren-Dean Langeveldt, Sebastian Link*

School of Information Management, The Victoria University of Wellington, Wellington, New Zealand

Abstract

Armstrong relations satisfy precisely those data dependencies that are implied by a given set of data dependencies. A common perception is that Armstrong relations are useful in the acquisition of data semantics, in particular since errors during the requirements elicitation have the most expensive consequences.

We report on some first empirical evidence for this perception regarding the class of functional dependencies (FDs). For this purpose, we investigate the usefulness of Armstrong relations with respect to various measures. Soundness measures how many of the as meaningful perceived FDs are actually meaningful. Completeness measures how many of the actually meaningful FDs are also perceived as meaningful.

Our experiment determines what and how much design teams learn about the application domain in addition to what they know prior to using Armstrong relations. The data analysis suggests that in using Armstrong relations it is not more likely to recognize meaningless FDs which are incorrectly perceived as meaningful, but it is more likely to recognize meaningful FDs that are incorrectly perceived as meaningless.

Our measures assess the quality of an FD set with respect to a target FD set, and therefore qualify naturally for the use in automated assessment tools, e.g. for database course exams or assignments.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

Armstrong relations are of interest in database theory and practice. Let $\Sigma \cup \{\varphi\}$ denote a set of functional dependencies (FDs). We say that $\Sigma$ implies $\varphi$, if every relation that satisfies every FD in $\Sigma$ also satisfies $\varphi$. That is, there is no counterexample relation that satisfies all FDs in $\Sigma$ and violates $\varphi$. We write $\Sigma \vdash \varphi$ to denote that $\Sigma$ implies $\varphi$ (and $\Sigma \not\vdash \varphi$ to denote that $\Sigma$ does not imply $\varphi$).

For a set $\Sigma$ of FDs, let $\Sigma^*$ denote the set of all FDs implied by $\Sigma$. For every FD $\varphi$ that is not in $\Sigma^*$, there is a counterexample relation $r_\varphi$ that satisfies all FDs in $\Sigma$ and violates $\varphi$. As a consequence of a result by Armstrong [1], there is a single counterexample relation that satisfies all FDs in $\Sigma^*$ and violates all FDs not in $\Sigma^*$. Following common terminology we call such a relation an Armstrong relation for $\Sigma$. The following example illustrates the potential benefits of utilizing Armstrong relations for the acquisition of meaningful FDs.

Let us assume that in developing an information system for some manufacturer of electrical goods we identify the processing of orders by retail sellers as a domain of interest. In particular, we define the relation schema ORDER that consists of the attributes Order#, Product#, Description, Qty and Total. These show for an order (identified by its order number Order#), a product in that order (identified by its unique product number Product#), a description Description of that product, the quantity Qty of that product in that order, and the total value Total (in some fixed currency) of that product in that order.
Suppose the designers of our information system have not been able yet to identify any meaningful FDs for the schema $\text{Order}$, i.e., $\Sigma = \emptyset$. Therefore, they decide to inspect a relation that faithfully represents the initial design draft of an empty FD set. The relation they decide to examine is the one in Table 1. This relation is Armstrong for the empty FD set $\Sigma$.

By inspecting the Armstrong relation the designers simply notice that the $\text{Oven}$ with Product# 521 is associated with the different quantities of 10 and 20 in the order with Order# 00724. This observation causes the design team to specify the FD $\text{Order}_\#, \text{Product}_\# \rightarrow \text{Qty}$ which states that the schema $\text{Order}$ records a unique quantity for the same product in the same order. A similar observation causes the design team to specify the FD $\text{Order}_\#, \text{Product}_\# \rightarrow \text{Total}$ which states that the order number and the product number together uniquely determine the total of the product in the order. Moreover, the design team observes that the product with Product# 521 has two different descriptions $\text{Microwave}$ and $\text{Oven}$. This observation causes the design team to ask the domain experts whether different descriptions can be associated with any product. Since the experts agree that this cannot be the case, the design team responds by specifying the FD $\text{Product}_\# \rightarrow \text{Description}$ which states that the description of a product is uniquely determined by the product number. We can see that, by inspecting the Armstrong relation above, the designers have successfully identified three meaningful FDs for the application domain. Furthermore, these three FDs together imply the FD $\text{Order}_\#, \text{Product}_\# \rightarrow \text{Description}, \text{Qty}, \text{Total}$.

Therefore, the design team recommends the attribute set ($\text{Order}_\#, \text{Product}_\#$) as a candidate key for the schema $\text{Order}$.

In general, a relation that satisfies an FD set $\Sigma$ but which is not Armstrong for $\Sigma$ will satisfy some FD that is not in $\Sigma^*$. Therefore, relations that are not Armstrong for a given FD set may not be able to reveal problems with the current design. For example, the relation in Table 2 is not Armstrong for the empty FD set $\Sigma$. While this relation satisfies $\Sigma$ (as every other relation does in this case), it gives the false impression that the current design, i.e. $\Sigma = \emptyset$, is acceptable. Specifically, the relation is not a faithful representation of the FD set $\Sigma$. For example, the relation does not violate the FD $\text{Order}_\#, \text{Product}_\# \rightarrow \text{Qty}$, nor the FD $\text{Order}_\#, \text{Product}_\# \rightarrow \text{Total}$, nor does it violate the FD $\text{Product}_\# \rightarrow \text{Description}$, even though they are not in $\Sigma^*$. Intuitively, an inspection of the relation in Table 2 does neither seem to encourage a design team to specify the FDs $\text{Order}_\#, \text{Product}_\# \rightarrow \text{Qty}$, $\text{Order}_\#, \text{Product}_\# \rightarrow \text{Total}$ nor does it seem to encourage the team to ask the domain experts whether different descriptions can be associated with the same product number.

This simple example illustrates the potential benefit of using Armstrong relations in the process of identifying the complete set of FDs that are meaningful for the underlying application domain. Failure to identify such a complete set means that the output of the requirements analysis is afflicted with errors.

Empirical studies show that more than half the errors which occur during systems development are requirements errors [2–4]. Requirements errors are also the most common cause of failure in systems development projects [2,5,6]. The cost of errors increases exponentially over the development life cycle: it is more than 100 times more costly to correct a defect post-implementation than it is to correct it during requirements analysis [7]. This suggests that it would be more effective to concentrate quality assurance efforts in the requirements analysis stage, in order to catch requirements errors as soon as they occur, or to prevent them from occurring altogether [8]. Hence, Armstrong relations appear to be a valuable tool for the requirements analysis of the target database. However, the question remains in what precise sense they are valuable.

Research gap and research questions: In previous work, Armstrong relations were called “user-friendly representations” of sets of data dependencies [9], and it was stated that they are “useful for database design” [9,10]. However, the phrase “useful for database design” was exclusively justified in terms of the structural and algorithmic properties of Armstrong relations. For instance, this may refer to the fact that FDs enjoy Armstrong relations, i.e., for every set $\Sigma$ of FDs there is an Armstrong relation for $\Sigma$. Note that it is everything but self-evident that a given class of data dependencies enjoys Armstrong relations [11]. Other interpretations of “useful” may refer to either the size of an Armstrong relation for an FD set $\Sigma$, e.g. the minimal number of tuples required for a relation to be Armstrong for $\Sigma$, or the existence/efficiency of algorithms to compute such an Armstrong relation. These

Table 1: An Armstrong relation for the empty FD set.

<table>
<thead>
<tr>
<th>Order#</th>
<th>Product#</th>
<th>Description</th>
<th>Qty</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>00723</td>
<td>389</td>
<td>Microwave</td>
<td>10</td>
<td>5000</td>
</tr>
<tr>
<td>00724</td>
<td>389</td>
<td>Microwave</td>
<td>10</td>
<td>5000</td>
</tr>
<tr>
<td>00724</td>
<td>521</td>
<td>Microwave</td>
<td>10</td>
<td>5000</td>
</tr>
<tr>
<td>00724</td>
<td>521</td>
<td>Oven</td>
<td>10</td>
<td>5000</td>
</tr>
<tr>
<td>00724</td>
<td>521</td>
<td>Oven</td>
<td>20</td>
<td>5000</td>
</tr>
<tr>
<td>00724</td>
<td>521</td>
<td>Oven</td>
<td>20</td>
<td>8000</td>
</tr>
</tbody>
</table>

Table 2: A relation not Armstrong for the empty FD set.

<table>
<thead>
<tr>
<th>Order#</th>
<th>Product#</th>
<th>Description</th>
<th>Qty</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>00723</td>
<td>389</td>
<td>Microwave</td>
<td>10</td>
<td>5000</td>
</tr>
<tr>
<td>00724</td>
<td>521</td>
<td>Oven</td>
<td>20</td>
<td>8000</td>
</tr>
</tbody>
</table>
interpretations of “useful” have received considerable interest from the research community, e.g. [9,12–14]. The authors are unaware of any research that provides evidence for the perception of usefulness of Armstrong relations in the requirements elicitation phase, as observed in the examples above. Bisbal and Grimson [15, p. 451] state that Armstrong relations are “expected to expose missing or undesirable functional dependencies”. So far, however, there is no empirical evidence that Armstrong relations really do assist design teams to decide whether a functional dependency is either meaningful or meaningless for the underlying application domain. In this paper, we address this research gap and seek answers to the following research questions:

1. In what precise sense can Armstrong relations be “useful” for the acquisition of meaningful functional dependencies for a given application domain?
2. Given a fixed and precise interpretation of the term “useful”, how “useful” are Armstrong relations for the acquisition of meaningful functional dependencies for a given application domain?

Note that question two subsumes the following question: given a fixed and precise interpretation of the term “useful”, are Armstrong relations “useful” for the acquisition of meaningful FDs for a given application domain. Throughout the paper, an FD is considered to be meaningful if every relation that represents a real-world instance over the given schema will satisfy the FD. This is different from an accidental FD [11], which is satisfied by some real-world instance (it is accidentally satisfied by this instance), but is violated by some other real-world instance. Therefore, the real problem for a database design team is the identification of meaningful FDs, and not the identification of accidental FDs. Note that an FD is perceived as meaningful for a relation schema by a design team that specified an FD set \( S \) if and only if the FD is satisfied by any Armstrong relation for \( S \). Intuitively, this points to the “usefulness” of Armstrong relations for identifying meaningful FDs.

**Research contributions:** In order to address our research questions we ask how the use of Armstrong relations can potentially contribute to the quality of the set of functional dependencies that a design team perceives as meaningful. For our analysis we measure this quality relative to a target set \( S \) of functional dependencies that forms a cover of all functional dependencies established as meaningful after consulting a group of domain experts.

In order to say something about the usefulness of Armstrong relations we measure what or how much design teams learn about the application domain in addition to what they already know prior to using Armstrong relations. Therefore, we first measure the quality without the use of an Armstrong relation, and then measure the quality after the use of Armstrong relations. If the quality increases by using Armstrong relations, then the Armstrong relations were indeed useful. Moreover, to measure the increase in quality (a negative increase is a decrease) we only compare the results from the same design team. Note that throughout the experiments the domain experts were present to answer potential questions from the design teams. For phase 1 we asked each design team to specify the set \( S_1 \) of FDs that they perceive as meaningful for the fixed underlying application domain, and then measured the quality of \( S_1 \) relative to the target set \( S \). For phase 2, we provided the same design team with an Armstrong relation for \( S_1 \), and asked them to revise \( S_1 \). After a number of repetitions of phase 2, the design team finalized their revised set \( S_2 \), and we then measured the quality of \( S_2 \) relative to the target set \( S \). If, on average, the quality of \( S_2 \) relative to \( S \) was better than the quality of \( S_1 \) relative to \( S \), then we concluded that Armstrong relations are useful for the acquisition of meaningful FDs with respect to the quality measure that we apply.

For our research questions we provide different measures of quality: proximity and minimality. Informally, proximity captures how close a set \( S \) is to the target set \( S \), and minimality captures the level of non-redundancy in the representation of \( S \). We further divide the measure of proximity into soundness and completeness. Soundness measures how many of the perceived meaningful FDs are actually meaningful, while completeness measures how many of the meaningful FDs were actually perceived as meaningful. Measures are defined with respect to the closure of attribute sets under implication. This guarantees that our measures are independent of the representation of the FD sets provided by the design teams. The usefulness of Armstrong relations is defined in terms of each of the measures and in two variations. For the first variation, we quantify usefulness as the arithmetic mean over the differences in quality (quality of \( S_2 \) minus quality of \( S_1 \)). Therefore, the first variation of usefulness measures the average gain in quality towards the target FD set. For the second variation, we quantify usefulness as the geometric mean over the ratios in quality (quality of \( S_2 \) divided by the quality of \( S_1 \)). Therefore, the second variation of usefulness measures the average growth in quality after using Armstrong relations.

According to our data analysis, we can report an average gain of 5% and an average growth by 7% in minimality after using Armstrong relations. The impact of Armstrong relations on the soundness of the FD sets is close to 0% in both gains and growths. By using Armstrong relations we gain on average 14% in terms of completeness and proximity on the target FD set, and the quality of the FD sets grows on average by 20% in completeness and proximity. A summary of the individual and average gains for each of the 20 design teams that participated in our project is illustrated in Fig. 1. Fig. 2 illustrates the individual and average growths in quality for the design teams.

Our first main result suggests that Armstrong relations help design teams to identify additional meaningful functional dependencies that were previously overlooked. In other words, by using an Armstrong relation it is more likely that meaningful FDs are recognized which were previously perceived as meaningless. The reason is that Armstrong relations violate FDs that are perceived as meaningless, and violations of actually meaningful FDs
are likely to be recognized. In this sense, our first main result empirically confirms Bisbal and Grimson’s expectation that Armstrong relation “expose missing functional dependencies”.

Our second main result suggests that Armstrong relations do not have an impact on the soundness of the FD sets (on average). In other words, by using an Armstrong relation it is not more likely to recognize an
Functional dependencies (FDs) are a fundamental concept in relational database theory. They represent the constraints that must be satisfied by the data in a database. Armstrong relations provide a partial solution to the problem of helping a design team think about whether some dependency is implied by the input, but simply notice whether the dependency is satisfied or violated by the relation. Noticeably, for the class of FDs this functionality is also included in the Database Design Expert System DBE [71], and for the class of standard FDs and inclusion dependencies such a functionality is also provided by the DBA companion [72]. All these approaches establish a partial solution to the problem of helping a design team think of what dependencies should be specified. In the present paper we establish some first quantitative and qualitative evidence to which degree Armstrong relations provide a partial solution to this problem. On the one hand, this provides further motivation for detailed studies of Armstrong relations and their properties. On the other hand, it also provides motivation to investigate the usefulness of Armstrong relations for.

New application areas involve data cleaning [55], data transformations [56], consistent query answering [57] and data exchange [33,58,59]. The effective and efficient utilization of FDs to this entire body of applications depends crucially on the correct acquisition of meaningful FDs from the underlying application domain. Failure to identify some of the meaningful FDs means that the various application areas cannot be explored completely.

In the remainder of this section we provide a brief overview about Armstrong databases. For a more complete and excellent summary we refer the interested reader to Fagin's treatment of the subject [60].

Armstrong [1] showed that for every set of FDs over a relation schema $R$ there is a single relation that satisfies precisely the FDs over $R$ that are in $\Sigma^*$. For this reason, Fagin [11] termed such relations Armstrong relations.

The question whether a class of data dependencies enjoys Armstrong relations can be intriguing. For example, the combined class of FDs and inclusion dependencies does not enjoy Armstrong databases in general [61]. However, if one does not permit so-called non-standard FDs (FDs that have an empty attribute set on their left-hand side), then the resulting combined class of standard FDs and inclusion dependencies does enjoy Armstrong databases [61]. Fagin has shown that the very general class of embedded implicational dependencies enjoys Armstrong relations [11]. Since then, the existence and properties of Armstrong databases have been investigated for many other classes of data dependencies. These include, but are not limited to, multivalued dependencies [62,63], degenerated multivalued dependencies [10], excluded functional dependencies [64], strong dependencies [65,66], partial dependencies [67], branching dependencies [68], key set dependencies [69], and cardinality constraints [13].

Since the focus of this paper is on the class of FDs the remainder of this section summarizes a few results on Armstrong relations established for this class. Silva and Mellkanoff [70] were the first to recognize the "practical" potential of Armstrong relations for the acquisition of meaningful dependencies. They implemented a tool that presents the design team with an Armstrong relation for a provided set of functional and multivalued dependencies. Their idea was that with the help of an Armstrong relation the design team did not have to think about whether some dependency is implied by the input, but simply notice whether the dependency is satisfied or violated by the relation. Noticeably, for the class of FDs this functionality is also included in the Database Design Expert System DBE [71], and for the class of standard FDs and inclusion dependencies such a functionality is also provided by the DBA companion [72]. All these approaches establish a partial solution to the problem of helping a design team think of what dependencies should be specified. In the present paper we establish some first quantitative and qualitative evidence to which degree Armstrong relations provide a partial solution to this problem. On the one hand, this provides further motivation for detailed studies of Armstrong relations and their properties. On the other hand, it also provides motivation to investigate the usefulness of Armstrong relations for.
the acquisition of meaningful data dependencies. In particular, we believe that it is a worthwhile task to develop a process model for the effective utilization of Armstrong relations in the acquisition process of data semantics.

So far, the investigation of Armstrong relations for the class of FDs was directed towards their structural and algorithmic properties. For example, Beeri, Dowd, Fagin and Statman characterized Armstrong relations for FD sets \( \Sigma \) in terms of attribute set closures under the implication of \( \Sigma \) [12]. They showed that a minimal-sized Armstrong relation can have exponentially many tuples in the number of attributes of the underlying relation schema [12]. This result is also important for the observations that we report in the present paper. In [12] the authors further show that the time complexity of finding an Armstrong relation is precisely exponential in the number of attributes [12]. However, considering that it can be more than 100 times more costly to correct a defect post-implementation than it is to correct it during requirements analysis [7], this is a small prize to pay.

Mannila and Räihä [9] state that Armstrong relations “can be used as user-friendly representations of dependency sets”, and therefore “should be useful to the database designer”. They provide an alternative characterization of Armstrong relations that they use to develop an output-sensitive algorithm to compute Armstrong relations of “small” size. Specifically, it is shown that the possibly exponential size of a minimal Armstrong relation is dependent on the number of FDs, and not on the number of attributes. They argue that in most situations that occur in practice, e.g. when a schema is normalized, the size is small enough for Armstrong relations to be useful in the database design process. Further, they exemplify the benefit of using both an abstract FD set and an Armstrong relation for that FD set in the design process of the target database. It is the aim of the present paper to give precise meaning to the phrase “useful in the database design process”.

Recently, the concept of Armstrong databases was extended to so-called informative Armstrong databases [14,73,74]. These are subsets of existing database relations that satisfy precisely the same set of FDs and inclusion dependencies that are satisfied by the entire relations in the database. In other words, they are dependency-preserving snapshots of a real database instance. As such, informative Armstrong databases can be used to verify whether the existing instance obeys all the database dependencies that the design team has in mind. Experiments have shown that the number of tuples in an informative Armstrong database is 0.6% of that in the existing database instance. In this sense, the size of informative Armstrong databases seems to indicate that they are indeed useful for a logical analysis of the existing database. In the spirit of our present paper, it would be interesting to investigate empirically in what precise sense the informative Armstrong relations are useful, e.g. in terms of the acquisition of additional functional and inclusion dependencies. We leave this direction for future work.

Armstrong databases can also be integrated into general approaches towards prototyping [73,75] and the sampling of test data [76–78]. Another approach that advocates the usefulness of example relations is given by Noble [79]. In [80] the authors discuss the usefulness of illustrative examples for understanding and developing algorithms in general. An alternative approach towards constraint acquisition in semantic data models is proposed in [81]. That approach focuses on the repeated inspection of certain two-element sample relations rather than Armstrong relations.

Recently, there have been efforts to improve the efficiency of database systems teaching [16,82,83]. By automating what can be replicated efficiently, course staff can devote their time to more important tasks such as individualized help or group discussions. Therefore, automation of assessments can lead to a higher productivity of the staff, improved objectivity in the assessments, higher student satisfaction through improved feedback, and the elimination of unnecessary costs and redundancy [16]. While this automation has been applied primarily to multiple choice questions, the measures that we introduce in our paper will allow to automatically assess the quality of FD sets put forward by students.

3. Preliminary definitions

We use this section to summarize the basic notions required for our treatment of functional dependencies and Armstrong relations.

3.1. Relations and functional dependencies

Let \( \forall = \{A_1, A_2, \ldots\} \) be a (countably) infinite set of distinct symbols, called attributes. A relation schema is a non-empty, finite subset \( R \) of \( \forall \) whose attributes represent column headers of a relation. Each attribute \( A \) of a relation schema \( R \) is associated with an infinite domain \( \text{dom}(A) \) which represents the set of possible values that can occur in the column named \( A \). If \( X \) and \( Y \) denote sets of attributes, then we follow a common convention in database terminology and write \( XY \) for \( X \cup Y \). If \( X = \{A_1, \ldots, A_n\} \), then we may write \( A_1 \cdots A_m \) for \( X \).

In particular, we may write simply \( A \) to represent the singleton \( \{A\} \). A tuple over \( R \) (\( R \)-tuple or simply tuple, if \( R \) is understood) is a function \( t : R \rightarrow \bigcup_{A \in \text{dom}(A)} \text{dom}(A) \) with \( t(A) \in \text{dom}(A) \) for all \( A \in R \). For \( X \subseteq R \) let \( t[X] \) denote the restriction of the tuple \( t \) over \( R \) to \( X \), and let \( \text{dom}(X) = \bigcap_{A \in X} \text{dom}(A) \) denote the Cartesian product of the domains of attributes in \( X \). A (finite) relation \( r \) over \( R \) is a (finite) set of tuples over \( R \).

Throughout the article we will use the following relation schema, denoted by \text{Schedule}. It consists of the four attributes \( C_{ID}, L_{Name}, Time \) and \( Room \) with implicit domain assignments. Intuitively, \text{Schedule} models weekly schedules in which a course with a (unique) course id \( C_{ID} \) is given by a lecturer with a (unique) name \( L_{Name} \) at a time \( Time \) (weekday and time) and in a room \( Room \) (building and room number). An example of a relation over \text{Schedule} is shown in Table 3. The column headers are the attributes of \text{Schedule}, and every row represents a tuple over \text{Schedule}. 

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|}
\hline
\text{ID} & \text{Name} & \text{Time} \\
\hline
1 & Smith & 9:00 AM
\end{tabular}
\caption{Example of a Schedule relation}
\end{table}
Functional dependencies between sets of attributes have always played a central role in the study of relational databases [21,22,84], and seem to be central for the study of database design in other data models as well [35,39–41,43,45,46,53,54].

A functional dependency (FD) over the relation schema R is an expression X → Y where X, Y ⊆ R. A relation r over R satisfies the functional dependency X → Y, denoted by r |= X → Y, if every pair of tuples in r that agrees on each of the attributes in X also agrees on each of the attributes in Y. That is, r |= X → Y if and only if for all t₁, t₂ ∈ r the following holds: if t₁[X] = t₂[X], then t₁[Y] = t₂[Y]. FDs of the form Y → Y are called non-standard, and functional dependencies that are not non-standard are called standard.

Consider the relation schema Schedule again. The experts of this domain may agree that no reasonable application domain should not be specified, i.e. it is meaningless for the course of a day’s lectures. This constraint is felt strong that there can never be two lecturers that can give the same course at the same time. This constraint is expressed as the FD

\[\text{Time, Room} \rightarrow C\text{ID}, \quad \text{C\text{ID}, Time} \rightarrow L\text{Name}.\]

The time constraint expresses this constraint. Moreover, the domain experts feel strong that there can never be two lecturers that can give the same course at the same time. This constraint is expressed as the FD

\[L\text{Name, Room} \rightarrow C\text{ID}.\]

The domain experts explain that it may well happen that different courses are taught by the same lecturer in the same room. Therefore, the FD

\[L\text{Name, Room} \rightarrow C\text{ID}\]

should not be specified, i.e. it is meaningless for the application domain Schedule.

For a set \(\Sigma\) of constraints over some relation schema R, we say that a relation \(r\) over \(R\) satisfies \(\Sigma\), denoted by \(r \models \Sigma\), if \(r\) satisfies every element of \(\Sigma\). If for some \(\sigma \in \Sigma\) the relation \(r\) does not satisfy \(\sigma\) we sometimes say that \(r\) violates \(\sigma\) (in which case \(r\) also violates \(\Sigma\)) and write \(r \not\models \sigma\) (\(r \not\models \Sigma\)).

### 3.2. Semantic implication and covers

For the design of a relational database schema dependencies are normally specified as semantic constraints on the relations which are intended to be instances of the schema. During the design process or the lifetime of a database one usually needs to determine further dependencies which are logically implied by the given ones. In line with the literature of database constraints, we restrict our attention to the implication of constraints in some fixed class \(C\). In this paper, we will consider the class of functional dependencies only.

Let \(R\) be a relation schema, and let \(\Sigma \cup \{\phi\}\) be a set of FDs over \(R\). We say that \(\Sigma\) (finely) implies \(\phi\), denoted by \(\Sigma \models \phi\), if and only if every (finite) relation \(r\) over \(R\) that satisfies \(\Sigma\) also satisfies \(\phi\). The (finite) implication problem is to decide, given any relation schema \(R\) and any set \(\Sigma \cup \{\phi\}\) of FDs over \(R\), whether \(\Sigma \models \phi\). If \(\Sigma\) does not (finely) imply \(\phi\) we may also write \(\Sigma \not\models \phi\). Note that for the class of FDs, sets \(\Sigma \cup \{\phi\}\) over a relation schema \(R\) are always finite. Moreover, finite and unrestricted implication problem coincide for the class of FDs. Hence, we will commonly speak of the implication problem.

For example, if \(\Sigma\) denotes the two FDs

\[\text{Time, Room} \rightarrow C\text{ID}, \quad \text{C\text{ID}, Time} \rightarrow L\text{Name},\]

then \(\Sigma\) implies the FD Time, Room → L_Name. That means, if a design team decides to specify the FD set \(\Sigma\) explicitly, then the team decides to implicitly specify the FD Time, Room → L_Name. In other words, there is no need to specify the FD Time, Room → L_Name explicitly as well. The FD set \(\Sigma\) above does not imply the FD L_Name, Time → Room, as the relation in Table 3 exemplifies. However, since no lecturer can be in multiple rooms at the same time, this FD expresses a meaningful semantic constraint that has not been captured implicitly by \(\Sigma\). Therefore, the FD needs to be specified explicitly as well.

For a set \(\Sigma\) of FDs over a relation schema \(R\), let \(\Sigma' = \{\phi | \Sigma \models \phi\}\) be its semantic closure, i.e., the set of all FDs over \(R\) implied by \(\Sigma\). We say that two FD sets \(\Sigma'\) and \(\Sigma''\) over \(R\) are equivalent, denoted by \(\Sigma' \equiv \Sigma''\), if \(\Sigma'\) implies every FD \(\sigma' \in \Sigma'\) and \(\Sigma''\) implies every FD \(\sigma \in \Sigma\). An FD set \(\Sigma'\) over \(R\) is said to be a cover for \(\Sigma\), if \(\Sigma'\) is equivalent to \(\Sigma\) [85]. Informally, a cover of an FD set is a different representation of the FD set. The set \(\Sigma\) is said to be non-redundant, if there is no FD \(\sigma \in \Sigma\) such that \((\neg \sigma) \cup \sigma\).

For example, the FD set \(\Sigma'\) consisting of

\[\text{Time, Room} \rightarrow C\text{ID}, \quad \text{C\text{ID}, Time} \rightarrow L\text{Name}\]

is equivalent to the FD set \(\Sigma\) consisting of

\[\text{Time, Room} \rightarrow C\text{ID}\] and \(\text{C\text{ID}, Time} \rightarrow L\text{Name}\).

While \(\Sigma\) is non-redundant, \(\Sigma'\) is not.

### 3.3. Attribute closures

For an attribute set \(X \subseteq R\) and a set \(\Sigma\) of FDs over \(R\), let \(X_\Sigma^+ = \{A \in R | \Sigma \models X \rightarrow A\}\) denote the attribute closure of \(X\) under \(\Sigma\) [21]. For an arbitrary set \(\Sigma \cup \{X \rightarrow A\}\) of FDs it is true that \(\Sigma\) implies \(X \rightarrow A\) if and only if \(A \in X_\Sigma^+\) [21]. If \(\Sigma\) is understood, we simply write \(X^+\) instead of \(X_\Sigma^+\). We call an attribute set \(X \subseteq R\) closed with respect to \(\Sigma\), if \(X = X^+_\Sigma\). It is easy to see that equivalent FD sets produce the same attribute closures. That is, if \(\Sigma\) is a cover of \(\Sigma'\), then for all attribute sets \(X \subseteq R\) it is true that \(X_\Sigma^+ = X_{\Sigma'}^+\). Hence, attribute closures are invariant under the representation of an FD set. Note that the above implication also holds in

<table>
<thead>
<tr>
<th>C_ID</th>
<th>L_Name</th>
<th>Time</th>
<th>Room</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALG100</td>
<td>Ullman</td>
<td>Fri, 04:00pm</td>
<td>Hunter 833</td>
</tr>
<tr>
<td>COMP320</td>
<td>Ullman</td>
<td>Fri, 04:00pm</td>
<td>Kirk 205</td>
</tr>
<tr>
<td>DB400</td>
<td>Ullman</td>
<td>Wed, 02:00pm</td>
<td>Kirk 205</td>
</tr>
<tr>
<td>DB400</td>
<td>Ullman</td>
<td>Wed, 02:00pm</td>
<td>MacKenzie 003</td>
</tr>
<tr>
<td>DB400</td>
<td>Ullman</td>
<td>Fri, 10:00am</td>
<td>MacKenzie 003</td>
</tr>
<tr>
<td>DB400</td>
<td>Widom</td>
<td>Tue, 03:00pm</td>
<td>MacKenzie 003</td>
</tr>
<tr>
<td>INFO230</td>
<td>Beeri</td>
<td>Tue, 03:00pm</td>
<td>Murphy 116</td>
</tr>
</tbody>
</table>
the opposite direction: if all attribute sets \(X \subseteq R\) satisfy the property that \(X'_C = X'_R\), then \(\Sigma\) and \(\Sigma'\) are equivalent, i.e., covers of one another.

For example, let \(\Sigma\) denote the FD set consisting of \(\text{Time, Room} \rightarrow \text{C_ID}\), and \(\text{C_ID, Time} \rightarrow \text{L_Name}\).

The attribute set \((\text{C_ID}, \text{Time}, \text{L_Name})\) is closed under \(\Sigma\), while the attribute set \((\text{C_ID}, \text{Time}, \text{Room})\) is not.

### 3.4. Armstrong relations

Let \(\Sigma\) be a set of constraints in a class \(C\) of data dependencies over some relation schema \(R\). A relation \(r\) over \(R\) is said to be an Armstrong relation for \(\Sigma\) under \(C\), if for all constraints \(\varphi\) in \(C\) over \(R\) it is true that \(r\) satisfies \(\varphi\) if and only if \(\varphi\) is implied by \(\Sigma\). Hence, \(r\) is an exact representation of the constraint set \(\Sigma\) in the sense that it satisfies all the constraints in \(\Sigma\) (and its implications), but violates all the constraints in \(\Sigma\) not implied by \(\Sigma\).

A class \(C\) of data dependencies is said to enjoy Armstrong relations if and only if for every relation schema \(R\), and for every set \(\Sigma\) of constraints in \(C\) over \(R\) there is some relation \(r\) over \(R\) that is an Armstrong relation for \(\Sigma\). Armstrong [1] showed implicitly that the class of FDs enjoys Armstrong relations. In general, the question whether a class of data dependencies enjoys Armstrong databases can be intriguing. For example, the combined class of FDs and inclusion dependencies does not enjoy Armstrong databases [61], but the combined class of standard FDs and inclusion dependencies does [61].

Given an Armstrong relation \(r\) for \(\Sigma\), the implication problem \(\Sigma = \varphi\) reduces to the problem whether \(r\) satisfies \(\varphi\).

For example, the relation \(r\) in Table 3 is Armstrong for the FD set \(\Sigma\) consisting of \(\text{Time, Room} \rightarrow \text{C_ID}\), and \(\text{C_ID, Time} \rightarrow \text{L_Name}\).

We can see that \(\Sigma\) does not imply the FD \(\text{L_Name, Time} \rightarrow \text{Room}\) since \(r\) violates this FD, and since \(r\) is Armstrong for \(\Sigma\) it follows that this FD is not in \(\Sigma^*\).

### 3.5. Acquisition of meaningful FDs

Before the process of designing the layout of the target database can start it is absolutely crucial to obtain a complete analysis of the requirements of the underlying application domain. Since future relations over our schema should represent only instances that actually occur in the application domain, we need a complete list of constraints that every such instance in the real world obeys. Suppose we call relations that represent the real world meaningful relations. Then we call a constraint meaningful, if it is satisfied by every meaningful relation, and meaningless otherwise. Since we want to exclude all those relations that are meaningless, we need to find a complete list of meaningful constraints. For the scope of this paper, we are interested in acquiring a cover of all meaningful FDs. Once we have such a cover, we can happily apply any of the tools that have been or will be developed for the purpose of database design. Note that these tools and the justification of these tools is based on having such a cover. That is, if we have not implicitly specified some meaningful FD, then the tools cannot achieve what the designers and end users are ultimately looking for.

The main objective of the paper is to provide some first empirical evidence of the usefulness of Armstrong relations for the acquisition of meaningful functional dependencies. More specifically, we seek answers to the following questions:

1. In what precise sense can Armstrong relations be “useful” for the acquisition of meaningful functional dependencies for a given application domain?
2. Given a fixed and precise interpretation of the term “useful”, how “useful” are Armstrong relations for the acquisition of meaningful functional dependencies for a given application domain?

### 4. Design of the experiment

In this section we describe the stages of our experiment, the general process for measuring the different notions of quality, the application domain and target FD set we utilize, the design teams and domain experts, and the limitations of our experiment. The specific quality measures are introduced in Section 5.

#### 4.1. The overall process of the experiment

Our data collection was conducted with the help of 20 design teams, each team had a unique integer id \(i\) between 1 and 20. Each design team was given the task to write down a cover for the set of all FDs that they perceive as meaningful for a fixed given application domain. The process was divided into two phases.

In the first phase, the design teams were asked to complete the given task with the help of a brief description of the intended semantics of the application domain, and by asking the present experts questions about the application domain. While these domain experts were happy to answer questions in natural language, they were unable to assist the design teams in their task to formally specify any of the FDs. At the end of the first phase design team \(i\) handed in their FD set \(\Sigma^*_1\).

At the beginning of the second phase, each design team \(i\) was given an Armstrong relation for the FD set \(\Sigma^*_1\). The design teams were instructed that this is a sample relation that is an exact representation of their FD set. That is, an FD is satisfied by the sample relation precisely when the team established this FD as meaningful for the application domain (the FD is implied by the FD set \(\Sigma^*_1\)). The design teams were then asked to revise their FD sets \(\Sigma^*_1\) with the additional help of the sample relation they were given. In particular, they were also able to ask the domain experts further questions about the application domain. Again, they did not receive any assistance in the specification of their FD set. During the second phase each design team had the opportunity to inspect Armstrong relations for
any of their revised FD sets. Eventually, each team handed in their revised sets $\Sigma_3^i$.

4.2. The overall process of measuring different notions of quality

In order to address our research questions we ask how the use of Armstrong relations can improve the quality of the FD set that a design team perceives as meaningful. That is, we measure the increase in quality (a decrease is a negative increase). In other words, we measure what or how much design teams learn about the application domain additionally to what they know prior to using Armstrong relations. The different quality measures we applied in this experiment are introduced in Section 5. For our analysis we measured the quality relative to a target FD set $\Sigma^t$ that forms a cover of all FDs established as meaningful after consultation with a group of domain experts. These were the exact domain experts that the design teams could ask question during the experiment.

In order to say something about the usefulness of Armstrong relations, we first measure the quality without the use of an Armstrong relation, and then measure the quality after the use of Armstrong relations. If the quality increases by using Armstrong relations, then the Armstrong relations were indeed useful. Moreover, to measure the increase in quality correctly, we only compare the results from the same design team.

More precisely, for each $i$ between 1 and 20 we measured the quality of $\Sigma_1^i$ relative to $\Sigma_1^t$, and the quality of $\Sigma_2^i$ relative to $\Sigma_2^t$. If, on average, the quality of $\Sigma_2^i$ relative to $\Sigma_2^t$ is better than the quality of $\Sigma_1^i$ relative to $\Sigma_1^t$, then we will conclude that Armstrong relations are useful for the acquisition of meaningful FDs with respect to the quality measure that we apply.

4.3. Application domain and target FD set

As application domain we chose the schema SCHEDULE, i.e., the attributes C_ID, L_Name, Time and Room with the intended semantics as outlined before. The authors of this article established a cover $\Sigma^t$ of the target set of all meaningful FDs over SCHEDULE. This cover consisted of the following FDs:

- L_Name, Time $\rightarrow$ Room,
- Time, Room $\rightarrow$ C_ID, and
- C_ID $\rightarrow$ L_Name.

The first FD L_Name, Time $\rightarrow$ Room indicates that the same lecturer cannot be in different rooms at the same time. The second FD Time, Room $\rightarrow$ C_ID says that no two different courses can be taught in the same room at the same time. For this FD to be specified the design teams may need to ask the domain experts whether there are any courses that can be co-taught with other courses at the same time in the same room. For our application domain this was not the case. Finally, the last FD C_ID $\rightarrow$ L_Name says that every course is taught by at most one lecturer. That is, lecturers do not co-teach a course. We chose to include this FD into our target set to demonstrate the necessity for design teams to interact with the domain experts. Design teams who fail to communicate with the domain experts and make the simple assumption that some course can be taught by multiple lecturers will therefore also fail to obtain a cover of the set of all meaningful FDs.

A representation-independent view of the target set $\Sigma^t$ are the attribute set closures of the relation schema SCHEDULE. These are given in Table 4, the closed attribute sets are in bold font. Note that we omitted the empty set $\emptyset$ as well as the full attribute subset that consisted of all attributes in SCHEDULE. The full attribute subset is closed under any constraint set, and is therefore independent of the FD sets that the design teams decide to specify. We also indicated to the design teams that there are no meaningful non-standard FDs (which have an empty attribute subset on their left-hand side). Therefore, the empty set was also closed, and none of the design teams decided to specify any non-standard FD.

4.4. The design teams and domain experts

We conducted the experiment with 50 third year students from two database classes. The students were familiar with concepts such as functional dependency, implied functional dependency, cover, third and Boyce-Codd normal form including their semantic justifications. Before the experiment, the students were not taught the concept of an Armstrong relation. For utilizing an Armstrong relation $r$ for $\Sigma$ effectively, the students were told that the relation $r$ they are given precisely represents their current FD set $\Sigma$. That is, $r$ satisfies an FD precisely if the design team currently perceives this FD as meaningful.

The students formed 20 different design teams with each team having either two or three members. Each design team was given the relation schema SCHEDULE together with a description of its intended semantics. The students had not encountered this application domain in their classes prior to the experiment. Moreover, they were told that the constraints that hold for this domain may or may not be different from the constraints that hold in the

<table>
<thead>
<tr>
<th>X $\subseteq$ SCHEDULE</th>
<th>Closure $X^t$ under $\Sigma^t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_ID</td>
<td>C_ID, L_Name</td>
</tr>
<tr>
<td>C_ID, L_Name</td>
<td>C_ID, L_Name</td>
</tr>
<tr>
<td>C_ID, Room</td>
<td>C_ID, L_Name, Room</td>
</tr>
<tr>
<td>C_ID, Time</td>
<td>C_ID, L_Name, Time</td>
</tr>
<tr>
<td>C_ID, Time, Room</td>
<td>C_ID, L_Name, Time, Room</td>
</tr>
<tr>
<td>C_ID, L_Name, Time</td>
<td>C_ID, L_Name, Time, Room</td>
</tr>
<tr>
<td>C_ID, L_Name, Room</td>
<td>C_ID, L_Name, Room</td>
</tr>
<tr>
<td>L_Name</td>
<td>L_Name</td>
</tr>
<tr>
<td>L_Name, Time</td>
<td>L_Name, Time, Room</td>
</tr>
<tr>
<td>L_Name, Room</td>
<td>L_Name, Time, Room</td>
</tr>
<tr>
<td>Time</td>
<td>Time</td>
</tr>
<tr>
<td>Room</td>
<td>Room</td>
</tr>
<tr>
<td>Time, Room</td>
<td>C_ID, L_Name, Time, Room</td>
</tr>
</tbody>
</table>
context of their own university. That is, to ensure they specify the correct constraints they should seek clarification with the domain experts.

The authors of this paper acted as domain experts for the application domain Schedule. That is, they were present during the experiments to clarify questions by the design teams. However, the students were not able to seek advice about the formal specification of their FD sets. For example, questions of the kind “Does this FD make sense?” were not answered.

Therefore, the authors of this paper acted in two roles: (i) establishing the target FD set which the quality of currently perceived meaningful FD sets can be measured against, and (ii) acting as domain experts. In practice, of course, there may not exist a unique target FD set simply because the body of knowledge provided to the design team does not allow to establish a unique FD set. For the sake of conducting the experiment, however, the presence of the domain experts and their consensus on the meaningful FDs guarantees that the quality of currently perceived meaningful FD sets can be measured in a transparent way.

4.5. Limitations

Some factors that may have influenced the results reported in this article include:

- students acting as database designers,
- familiarity of students with application domain,
- number and size of the application domain,
- time constraints,
- assumption that domain experts are present, and
- assumption that there is consensus among domain experts.

The argument that students are not database designers is certainly a valid one, in particular with respect to experience and their communication skills with the domain experts. However, it is by no means obvious that a database designer is more skilled or motivated to identify meaningful FDs. For example, the experience of a designer could suggest that it is only worth to identify meaningful keys (or even just the primary key) [25]. Therefore, experienced designers may not have an advantage after all in the task of identifying a cover of all meaningful FDs. The lack of experience in students might contribute to being more open-minded or motivated to complete the given task. Also, if Armstrong relations are already helpful for inexperienced students, then they are certainly helpful to experienced database designers. For these reasons we found that students with a solid education in database concepts represent a reasonable choice for the design teams.

A concern might be raised about the database designers’ level of familiarity with the application domain. In practice, however, it may also be the case that designers are chosen that have some knowledge about the domain of interest. Of course, this is not always possible. Moreover, some of the FDs that we included in the target set model constraints that are quite different from what the students may have experienced at their university. It was pointed out to them that they cannot make assumptions about the domain, and that it is safer to consult with the domain experts. We acknowledge that the familiarity with the application domain might have made it easier for some of the students to ask the experts the right questions. This, however, balances nicely the time constraints imposed on our experiment, and also the students’ lack of experience as database designers. The results confirm that the application domain was not too simple: design teams perceived some of the meaningless FDs as meaningful and some of the meaningful FDs as meaningless. Finally, let us suppose that the degree of familiarity was higher than in a “real-world” situation. Consequently, with a normal degree of familiarity the design teams would have specified a bigger number of meaningless FDs and a smaller number of meaningful FDs prior to inspecting an Armstrong relation. Consequently, the impact of Armstrong relations should be even bigger than we report here. Intuitively, this makes sense: the less one knows about the application domain, the more assistance a good sample relation provides.

Another concern is that we only consider one application domain, and this domain has only four attributes. We encourage to conduct the experiments for different application domains. Our simple experiment is illustrative enough to show how our measures can be applied to collect more evidence. The specific results and percentages that we report for our experiment are likely to vary for different domains. However, we strongly believe that the conclusion and overall insights will remain the same. We are confident that the first empirical evidence that we provide in this article is rather intuitive and may therefore convince designers to appreciate Armstrong relations and apply them effectively during the acquisition process. Regarding the size of our application, we believe that the results we report are already interesting for such a small number of attributes, and that the impact of Armstrong relations will be even more positive for schemas with more attributes. The reason is that the number of FDs to consider grows exponentially with the number of attributes. Therefore, it becomes more difficult to discover meaningful FDs and eliminate meaningless FDs.

The experiment was conducted under stricter time constraints than in practice. However, the complexity of the underlying application domain gave the design teams sufficient time for consulting the domain experts, discuss their understandings and formulate their solutions.

In practice, it happens quite often that domain experts are unavailable or non-existent. We believe that in these cases Armstrong relations are even more useful. The reason is, as we will see later, that Armstrong relations can pinpoint the decision whether an FD is meaningful or not. When in doubt it is likely that the designer will not specify an FD in order to guarantee that meaningful relations can still become database instances. That is, Armstrong relations may prevent the database designer to make assumptions about the application domain.

Furthermore, it is most likely the average case that there is no complete consensus among the domain experts.
experts, in particular not about the meaningfulness of some constraint. Here we believe that Armstrong relations can help to pinpoint the inconsistencies between the domain experts’ opinions. A more complex experiment could leave the meaningfulness of some of the possible FDs open. In that case, several target FD sets would co-exist, and the measures would have to be generalized. One possibility would be to measure the quality of an FD set relatively to all possible target FD sets, and then to take an optimum. This issue, however, was beyond the scope of this first study.

5. Quality measures

In this section we introduce the formal definitions of our four quality measures. These include the soundness and completeness of an FD set relative to a target FD set as well as the proximity of two FD sets. Finally, we use the minimality of an FD set to measure the quality of the representation of an FD set. The measures of soundness, completeness and proximity have natural complements: unsoundness, incompleteness and distance, respectively. In particular, as will we see, distance defines a metric on the set of all FD sets modulo equivalence. However, since we want to test the usefulness of Armstrong relations, i.e., whether there is a quality increase in using Armstrong relations, we utilize primarily the measures of soundness, completeness and proximity.

Throughout this section let \( P_0(R) = P(R) - \{0, R\} \) denote the set of all attribute subsets of some relation schema \( R \) without the empty and full attribute subset. Moreover, we use FD sets \( \Sigma \) and \( \Sigma' \) over \( R \), and assume that \( R \) contains at least two attributes, i.e., \( |R| > 1 \).

5.1. Proximity and distance

Informally, proximity is intended to measure the “closeness” between two FD sets. In terms of our experiment, phase 1 establishes for each individual design team the proximity of their perceptions of meaningful FDs to the target FD set prior to using an Armstrong relation. Phase 2 measures for each individual design team the proximity of their revised perceptions to the target FD set after using Armstrong relations. Therefore, the motivation of proximity is to measure how much better we can approximate the target FD set when Armstrong relations are utilized.

The proximity between two FD sets \( \Sigma \) and \( \Sigma' \) is the collection of all attribute subsets that have matching closures \( X_\Sigma \) and \( X_{\Sigma'} \). More formally, we obtain

\[
prox(\Sigma, \Sigma') = \{X \in P_0(R) | X_\Sigma = X_{\Sigma'}\}.
\]

Note that

\[
dist(\Sigma, \Sigma') = P_0(R) - prox(\Sigma, \Sigma')
\]

satisfies the following properties:

1. \( dist(\Sigma, \Sigma') = 0 \) if and only if \( \Sigma \equiv \Sigma' \).
2. \( dist(\Sigma, \Sigma') = dist(\Sigma', \Sigma) \), and
3. \( dist(\Sigma, \Sigma') \subseteq dist(\Sigma, \Sigma') \cup dist(\Sigma', \Sigma') \).

The third property is satisfied since a set \( X \in P_0(R) \) that satisfies \( X_\Sigma \neq X_{\Sigma'} \) must necessarily satisfy \( X_\Sigma \neq X_{\Sigma'} \) or \( X_{\Sigma'} \neq X_{\Sigma'} \).

Consequently, the function \( \text{dist}(\Sigma, \Sigma') \), that maps the FD sets \( \Sigma \) and \( \Sigma' \) to the cardinality of the set \( \text{dist}(\Sigma, \Sigma') \), defines a metric on the set of all FD set pairs over some fixed relation schema modulo equivalence.

To quantify the degree of proximity between two FD sets we could simply take the cardinality of the set \( prox(\Sigma, \Sigma') \). This, however, does not illustrate convincingly how many out of all attribute subsets have the same closure. This is achieved by the following ratio:

\[
prox\text{-ratio}(\Sigma, \Sigma') := \frac{|prox(\Sigma, \Sigma')|}{|P_0(R)|}.
\]

Note that the ratio is well-defined since we assume that the underlying relation schema contains at least two attributes.

5.2. Soundness and unsoundness

Informally, soundness measures which of the FDs perceived as meaningful by the design teams are actually meaningful (i.e. implied by the target set). Therefore, the motivation of soundness is to measure which and how many of the FDs, that are incorrectly perceived as meaningful by the design teams, can be recognized as meaningless with the help of Armstrong relations.

The soundness of an FD set \( \Sigma \) relative to an FD set \( \Sigma' \) is the collection of all those attribute subsets whose closure with respect to \( \Sigma \) is contained in the closure with respect to \( \Sigma' \). More formally, we obtain

\[
sound_{\Sigma'}(\Sigma) := \{X \in P_0(R) | X_\Sigma \subseteq X_{\Sigma'}\}.
\]

Naturally, the complement of \( sound_{\Sigma'}(\Sigma) \) records all those attribute subsets whose closure with respect to \( \Sigma \) contains attributes not contained in the closure with respect to \( \Sigma' \):

\[
unsound_{\Sigma'}(\Sigma) := P_0(R) - sound_{\Sigma'}(\Sigma).
\]

The corresponding ratio

\[
sound\text{-ratio}_{\Sigma'}(\Sigma) := \frac{|sound_{\Sigma'}(\Sigma)|}{|P_0(R)|}
\]

illustrates how many out of all attribute subsets have sound-\( \Sigma' \)- closures relatively to \( \Sigma' \).

5.3. Completeness and incompleteness

Informally, completeness measures which of the meaningful FDs (i.e. those implied by the target set) are actually perceived as meaningful by the design teams. Therefore, the motivation of completeness is to measure which and how many of the FDs, that are incorrectly perceived as meaningless by the design teams, can be recognized as meaningful with the help of Armstrong relations.

The completeness of an FD set \( \Sigma \) relative to an FD set \( \Sigma' \) is the collection of all those attribute subsets whose closure with respect to \( \Sigma' \) is contained in the closure with
respect to $\Sigma$. More formally, we obtain:
\[ \text{complete}_\Sigma(\Sigma) := \{X \in P_0(R) | X \subseteq X_0^\Sigma \}. \]

Naturally, the complement of $\text{complete}_\Sigma(\Sigma)$ records all those attribute subsets whose closure with respect to $\Sigma$ contains attributes not contained in the closure with respect to $\Sigma$:
\[ \text{incomplete}_\Sigma(\Sigma) := P_0(R) - \text{complete}_\Sigma(\Sigma). \]

The corresponding ratio
\[ \text{complete-ratio}_\Sigma(\Sigma) := \frac{|\text{complete}_\Sigma(\Sigma)|}{|P_0(R)|} \]
illuminates how many out of all attribute subsets have complete $\Sigma$- closures relatively to $\Sigma$.

5.4. Minimality

This measure was simply included for reasons of curiosity. It shows the efficiency of the design teams in representing their FD sets. The literature suggests various measures for minimality, e.g. non-redundant, canonical, minimum and optimal covers [85]. For the purpose of this study we were happy with the simplest of these forms, i.e., non-redundant covers. Let $\text{nrc}(\Sigma) = \{\Sigma' \subseteq \Sigma | \Sigma' \equiv \Sigma \land \Sigma' \text{ is non-redundant}\}$ denote the set of all non-redundant covers of $\Sigma$ contained in $\Sigma$. Let further $\min(\Sigma)$ denote the set
\[ \{\Sigma' \in \text{nrc}(\Sigma) | \Sigma'/nrc(\Sigma)| < |\Sigma''/nrc(\Sigma)|\} \]
i.e., the elements of $\text{nrc}(\Sigma)$ of maximum cardinality. Finally, let
\[ \text{min-ratio}(\Sigma) = \frac{|X|}{|\Sigma|} \text{ where } X \in \min(\Sigma). \]

A summary of the different measures that we apply is given in Fig. 3. In particular, we have that
\[ P_0(R) = \text{dist}(\Sigma, \Sigma^\Sigma) \cup \text{prox}(\Sigma, \Sigma^\Sigma), \]
and
\[ \text{prox}(\Sigma, \Sigma^\Sigma) = \text{sound}_\Sigma(\Sigma) \cap \text{complete}_\Sigma(\Sigma) \]
hold for any FD sets $\Sigma$ and $\Sigma^\Sigma$ over relation schema $R$.

5.5. Notions of usefulness

We will now define the quantity of usefulness in terms of each of our four quality measures. For each of the measures, we consider two variations of usefulness. The first variation determines usefulness as the average gain in quality of the FD sets relative to the target set. The second variation determines usefulness as the average growth of quality of the FD sets. Naturally, the first variation utilizes the notion of an arithmetic mean while the second variation utilizes the notion of a geometric mean.

5.5.1. Gains

For every measure in $\{\text{sound}, \text{complete}, \text{min}\}$ and the number $n$ of design teams in our experiment we define gain-in-measure, for $1 \leq i \leq n$ as
\[ \text{measure-ratio}_{\Sigma^i}(\Sigma^i) - \text{measure-ratio}_{\Sigma_1}(\Sigma_1) \]
and gain-in-measure as the arithmetic mean over these differences:
\[ \frac{\sum_{i=1}^{n} \text{gain-in-measure}_i}{n}. \]

Apart from the difference in notation, the same definition applies to the measure of proximity, i.e., gain-in-prox, for $1 \leq i \leq n$ is defined as
\[ \text{prox-ratio}_{\Sigma^i, \Sigma^i} - \text{prox-ratio}_{\Sigma_1, \Sigma_1} \]
and gain-in-prox is defined as
\[ \frac{\sum_{i=1}^{n} \text{gain-in-prox}_i}{n}. \]

For every measure in $\{\text{sound}, \text{complete}, \text{prox}, \text{min}\}$ we say that the use of Armstrong relations for the acquisition of meaningful FDs achieved a gain of $\text{gain-in-measure} \times 100\%$ in terms of the measure. We say that Armstrong relations are useful for the acquisition of meaningful FDs in terms of the gain of the measure, if the gain is positive.

5.5.2. Growths

For every measure in $\{\text{sound}, \text{complete}, \text{min}\}$ and the number $n$ of design teams in our experiment we define growth-in-measure, for $1 \leq i \leq n$ as
\[ \text{measure-ratio}_{\Sigma^i}(\Sigma^i) \]
and growth-in-measure as the geometric mean over these ratios:
\[ \sqrt[n]{\prod_{i=1}^{n} \text{growth-in-measure}_i}. \]

Apart from the difference in notation, the same definition applies to the measure of proximity, i.e., gain-in-prox, for $1 \leq i \leq n$ is defined as
\[ \text{prox-ratio}_{\Sigma^i, \Sigma^i} \]
and gain-in-prox is defined as
\[ \sqrt[n]{\prod_{i=1}^{n} \text{growth-in-prox}_i}. \]
For every measure in [sound, complete, prox, min] we say that the use of Armstrong relations for the acquisition of meaningful FDs achieved a growth by (growth-in-measure – 1) × 100% in terms of the measure. We say that Armstrong relations are useful for the acquisition of meaningful FDs in terms of the growth of the measure, if the growth growth-in-measure is bigger than one.

6. Data collection—a sample

In this section we illustrate a sample run of our experiment based on the solutions that design team 3 provided.

6.1. Instructions

The design teams were given the relation schema Schedule with the four attributes

C_ID, L_Name, Time and Room.

Further, they were given the intended meaning of the relation schema. That is, Schedule collects information about the weekly schedule of university courses in a semester. More precisely, a university course with a unique course id C_ID (e.g. INFO341) is given by a lecturer with a unique name L_Name (e.g. Tieong Goh) at a time (weekday plus time, e.g. Thu, 02:10pm) in a room (building plus room number, e.g. Hunter 113).

Design teams were given the task to write down a cover for the set of all functional dependencies that they perceive as meaningful for the schema Schedule. The teams were encouraged to ask the present domain experts questions about the underlying application domain. However, they were also instructed that the domain experts were unable to provide any assistance in the specification of the functional dependencies.

6.2. Phase 1

Design team 3 submitted the following set \( \Sigma_1 \) of FDs:

- \( L_Name, Time \rightarrow Room \),
- \( C_ID, Time, Room \rightarrow L_Name \),
- \( L_Name \rightarrow C_ID \), and
- \( Time, Room \rightarrow C_ID \).

The representation of \( \Sigma_1 \) as attribute closures is illustrated in Table 5.

For the soundness of \( \Sigma_1 \) relative to \( \Sigma^t \) we need to inspect every \( X \in P_0(Schedule) \) whether \( X_{3}^t \subseteq X_{1}^t \), holds. Comparing Table 5 with Table 4 we obtain \( Sound_{\Sigma_1}^t(\Sigma_1^t) = \frac{14}{14} = 0.86 \), that is \( \Sigma_1^t \) was 86% sound relative to \( \Sigma^t \).

The proximity-ratio of \( \Sigma_1 \) relative to \( \Sigma^t \) is given by

\[
prox_{\Sigma_1}(\Sigma_1^t, \Sigma^t) = \frac{9}{14} = 0.64.
\]

that is the proximity between \( \Sigma_1 \) and \( \Sigma^t \) was 64%.

The FD set \( \Sigma_1^t \) was non-redundant, therefore

\[
min_{\Sigma_1}(\Sigma_1^t) = 1.
\]

6.3. Armstrong relation

The design team 3 was then given the Armstrong relation for \( \Sigma_1^t \) that is illustrated in Table 6. The instructions from Phase 1 also applied to Phase 2, and each team was asked to revise their FD sets based on the different Armstrong relations they were given for their FD sets \( \Sigma_1 \) from Phase 1.

6.4. Phase 2

At the end of Phase 2, design team 3 specified the set \( \Sigma_2^t \):

- \( L_Name, Time \rightarrow C_ID \),
- \( C_ID, Time \rightarrow Room \).

The proximity-ratio of \( \Sigma_2^t \) relative to \( \Sigma_1^t \) is given by

\[
prox_{\Sigma_2}(\Sigma_2^t, \Sigma_1^t) = \frac{9}{14} = 0.64.
\]

that is the proximity between \( \Sigma_2^t \) and \( \Sigma_1^t \) was 64%.

The FD set \( \Sigma_2^t \) was non-redundant, therefore

\[
min_{\Sigma_2}(\Sigma_2^t) = 1.
\]
Table 6
An Armstrong relation for $\Sigma^1$.

<table>
<thead>
<tr>
<th>C_ID</th>
<th>L_Name</th>
<th>Time</th>
<th>Room</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB343</td>
<td>Mary Tate</td>
<td>Thu, 09:10am</td>
<td>Hunter 833</td>
</tr>
<tr>
<td>DB343</td>
<td>Mary Tate</td>
<td>Tue, 10:10am</td>
<td>Kirk 205</td>
</tr>
<tr>
<td>DB343</td>
<td>Mary Tate</td>
<td>Wed, 11:00am</td>
<td>Kirk 205</td>
</tr>
<tr>
<td>DB343</td>
<td>Alex Potanin</td>
<td>Wed, 11:00am</td>
<td>MacKenzie 003</td>
</tr>
<tr>
<td>DB343</td>
<td>Sid Huff</td>
<td>Tue, 01:00pm</td>
<td>MacKenzie 003</td>
</tr>
<tr>
<td>DB220</td>
<td>James Noble</td>
<td>Tue, 01:00pm</td>
<td>Murphy 116</td>
</tr>
<tr>
<td>INFO241</td>
<td>Tiong Goh</td>
<td>Mon, 09:00am</td>
<td>Murphy 116</td>
</tr>
</tbody>
</table>

- Time, Room $\rightarrow$ C_ID, and
- C_ID $\rightarrow$ L_Name.

The FD set $\Sigma_2^2$ is a non-redundant cover of $\Sigma^1$. Therefore, we have

- $\text{sound-ratio}_2(\Sigma_2^2) = P_0(\text{Schedule})$,
- $\text{complete-ratio}_2(\Sigma_2^2) = P_0(\text{Schedule})$, and
- $\text{prox-ratio}_2(\Sigma_2^2) = P_0(\text{Schedule})$.

Furthermore,

- $\text{sound-ratio}_2(\Sigma_2^2) = 1$,
- $\text{complete-ratio}_2(\Sigma_2^2) = 1$,
- $\text{prox-ratio}_2(\Sigma_2^2, \Sigma^1) = 1$, and
- $\text{min-ratio}(\Sigma_2^2) = 1$.

This corresponds to a 14% gain in terms of soundness, a 21% gain with respect to completeness, and a 36% gain in terms of proximity. Moreover, it means a 16% growth in soundness, a 27% growth in completeness and a 56% growth in proximity.

6.5. Automated assessment and feedback for course work

It seems relatively natural to use our measures for automating the assessment of course work submitted by students.

Functional dependencies are taught in most undergraduate database courses. It is a worthwhile experience for students to learn about the difficulties in identifying the meaningful functional dependencies for an application domain. However, it can be tedious to mark such assignments since the solutions will usually be different from student to student. The manual marking is also susceptible to errors.

Our experiment, and in particular our measures, offer a convenient way to provide an automated assessment. Moreover, Armstrong relations can be used to provide automated feedback to the students about their solutions.

In general, the course instructor can pick an arbitrary relation schema and fix a target FD set $\Sigma^i$. According to the target set, a natural language description of the expected FDs should be provided to the students. If this is to be made more realistic, then instructors may choose to withhold some of the information and only offer additional explanations when the students ask the right questions about the domain. This mimics the real-world case with the consultation of domain experts and can be modeled by on-line discussion forums, for example.

Whenever a student submits an FD set $\Sigma$ our measures can be computed relatively to the target FD set $\Sigma^i$. Qualitative feedback can be provided in terms of our measures, and a quantitative assessment can be made by using the corresponding ratios. A course-dependent assignment of marks can be defined relatively to the percentages achieved by the students. The course instructor can also choose how refined the individual feedback for the students may be. It seems natural to use proximity as a measure to assess how well the students did, but this can be broken down into soundness and completeness in the feedback. Armstrong relations can also provide an excellent individual feedback to the students, either for their submitted solutions or as an additional help that they can access on-line.

7. Quantitative data analysis

In this section, we present and analyze the different ratios that the design teams achieved during our experiment.

7.1. Soundness

The soundness ratios of the FD sets $\Sigma_1^i$ and $\Sigma_2^i$ relative to the target set $\Sigma^i$ are recorded in Table 7 and illustrated in Fig. 4.

Table 7 shows that $\text{gain-in-sound}$ in percent was $-0.45$, and $\text{growth-in-sound}$ in percent was $-0.58$. Therefore, on average, there was no increase and no decrease in terms of soundness, if we speak loosely. According to our terminology, Armstrong relations are not useful for the acquisition of meaningful FDs in terms of gain and growth in soundness.

It means that by inspecting an Armstrong relation, on average, (i) it is unlikely that further meaningless FDs are incorrectly added and (ii) it is unlikely that meaningless FDs are correctly removed. Property (i) is intuitive: if an FD is already correctly perceived as meaningless, then an Armstrong relation should give further evidence that the FD is meaningless by violating the FD. We will now argue that property (ii) is also intuitive.

In fact, an FD that is perceived as meaningful by a design team $i$ is implied by $\Sigma_1^i$, and therefore satisfied by any Armstrong relation for $\Sigma_1^i$. For the design team members to notice that this FD is actually meaningless, they have to recognize that the satisfaction of that FD by the given Armstrong relation does not make sense. For this to happen, they need to recognize that the FD is satisfied by the Armstrong relation and that this does not make sense for the application domain. Even worse, to recognize that the FD is satisfied by the Armstrong relation all pairs of distinct tuples need to be examined. In this case, the potential necessity of having a number of tuples in the Armstrong relation that is exponential in the number of attributes makes this endeavor even less likely. Additionally, the satisfaction of an FD $X \rightarrow Y$ cannot always
be explicit, i.e., there may only be tuples which already disagree on the attributes in $X$ (for instance when $X$ is a key, i.e., the FD $X \rightarrow R$ is implied by $\Sigma_1^i$). In summary, it is not likely that a design team simply notices that some FD is satisfied by the Armstrong relation, and then concludes that this does not make sense in the context of this application domain. Therefore, meaningless FDs that are incorrectly perceived as being meaningful prior to inspecting Armstrong relations are also likely to be incorrectly perceived as being meaningful after inspecting Armstrong relations. Indeed, our results suggest this empirically.

Another observation is that exactly half of the design teams ($11 \leq i \leq 20$) only specified meaningful FDs. In

<table>
<thead>
<tr>
<th>Team $i$</th>
<th>$\text{sound-ratio}_i(\Sigma_1^i)$ in percent</th>
<th>$\text{sound-ratio}_i(\Sigma_1^S)$ in percent</th>
<th>$\text{gain-in-sound}_i$ in percent</th>
<th>$\text{growth-in-sound}_i$ in percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>93</td>
<td>71</td>
<td>-22</td>
<td>-24</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>86</td>
<td>-14</td>
<td>-14</td>
</tr>
<tr>
<td>3</td>
<td>86</td>
<td>14</td>
<td>-7</td>
<td>-7</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>93</td>
<td>14</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>86</td>
<td>100</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>93</td>
<td>64</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>86</td>
<td>71</td>
<td>-15</td>
<td>-17</td>
</tr>
<tr>
<td>8</td>
<td>57</td>
<td>36</td>
<td>36</td>
<td>63</td>
</tr>
<tr>
<td>9</td>
<td>86</td>
<td>64</td>
<td>-22</td>
<td>-25</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 7
Soundness analysis.

**Fig. 4.** Soundness before and after using Armstrong relations.
other words, half of the design teams did not incorrectly specify any meaningless FDs. This means also that gain-in-sound, and growth-in-sound, amount to 0% for i = 10, . . . , 20.

Five of the remaining 10 design teams (1 ≤ i ≤ 10) had a decrease in terms of soundness, i.e., gain-in-sound, < 0 and growth-in-sound, < 1 for i ∈ {1, 2, 4, 8, 10}, and four teams had an increase, i.e., gain-in-sound, > 0 and growth-in-sound, > 1 for i ∈ {3, 5, 6, 9}. Out of the five design teams for which we recorded a decrease, two (i = 2 and 4) had specified only meaningful FDs in Σ^i 1. Out of the four design teams for which we recorded an increase, three (i = 3, 5 and 6) correctly removed all of the meaningless FDs from Σ^i 1.

The maximum for gain-in-sound, was recorded for design team i = 9 which reached 36%. The minimum for gain-in-sound, was recorded for design teams i = 1 and 10 which both had a 22% loss.

The maximum for growth-in-sound, was recorded for design team i = 9 which reached 63%. The minimum for growth-in-sound, was recorded for design team i = 10 which had a 25% loss.

### 7.2. Completeness

The completeness ratios of the FD sets Σ^i 1 and Σ^i 2 relative to the target set Σ 1 are recorded in Table 8 and illustrated in Fig. 5.

Table 8 shows that gain-in-complete in percent was 14.35 and growth-in-complete in percent was 20.19. According to our terminology, the use of Armstrong relations for the acquisition of meaningful FDs achieved a gain of 14% and a growth by 20% in terms of completeness.

It means that by inspecting an Armstrong relation, on average, (i) it is likely that further meaningful FDs are correctly discovered and (ii) it is unlikely that meaningful FDs are incorrectly removed. Property (ii) is intuitive: if an FD is already correctly perceived as meaningful, then an Armstrong relation should give further evidence that the FD is meaningful by satisfying the FD. We will now argue that property (i) is also intuitive.

In fact, an FD that is not perceived as meaningful by a design team i is not implied by Σ^i 1, and therefore violated by any Armstrong relation for Σ^i 1. For the design team members to notice that this FD is actually meaningful, they simply need to recognize that the violation of this meaningful FD by the Armstrong relation does not make sense. For this to happen, they simply need to find (any) two distinct tuples that agree on the left-hand side of the FD but disagree on some attribute of the right-hand side, and notice that this does not make sense in the context of the application domain. If the FD is indeed meaningful, then this should be relatively easy to recognize. Here, the smaller the number of tuples in the Armstrong relation is, the easier it should be to notice such a violation. Therefore, meaningful FDs that are incorrectly perceived as meaningless prior to the inspection of an Armstrong relation are likely to be correctly perceived as meaningful FDs after the inspection of Armstrong relations. Indeed, our results suggest this empirically.

Another important observation is that apart from one exception (i = 9) we did not record any decrease in completeness, i.e., gain-in-complete, ≥ 0 and growth-in-complete, ≥ 1 for all i ∈ {1, . . . , 20} – {19} (95% of all design teams). Three out of these 19 design teams (i = 12, 13 and 19) already achieved 100% completeness in phase 1. Moreover, 14 design teams had an actual increase in completeness, i.e., gain-in-complete, > 0 and growth-in-complete, > 1 for 70% of all design teams.

The maximum for gain-in-complete, was recorded for design team i = 15 which gained 43%. The minimum for

<table>
<thead>
<tr>
<th>Team i</th>
<th>complete-ratio_{Σ 1^i} in percent</th>
<th>complete-ratio_{Σ 2^i} in percent</th>
<th>gain-in-complete, in percent</th>
<th>growth-in-complete, in percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>64</td>
<td>79</td>
<td>15</td>
<td>23</td>
</tr>
<tr>
<td>2</td>
<td>79</td>
<td>93</td>
<td>14</td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>79</td>
<td>100</td>
<td>21</td>
<td>27</td>
</tr>
<tr>
<td>4</td>
<td>86</td>
<td>100</td>
<td>36</td>
<td>56</td>
</tr>
<tr>
<td>5</td>
<td>64</td>
<td>86</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>93</td>
<td>86</td>
<td>22</td>
<td>34</td>
</tr>
<tr>
<td>7</td>
<td>86</td>
<td>100</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>100</td>
<td>86</td>
<td>−14</td>
<td>−14</td>
</tr>
<tr>
<td>9</td>
<td>100</td>
<td>93</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>86</td>
<td>71</td>
<td>22</td>
<td>31</td>
</tr>
<tr>
<td>11</td>
<td>100</td>
<td>93</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>93</td>
<td>64</td>
<td>29</td>
<td>45</td>
</tr>
<tr>
<td>15</td>
<td>43</td>
<td>43</td>
<td>43</td>
<td>100</td>
</tr>
<tr>
<td>16</td>
<td>93</td>
<td>100</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>17</td>
<td>64</td>
<td>100</td>
<td>36</td>
<td>56</td>
</tr>
<tr>
<td>18</td>
<td>79</td>
<td>93</td>
<td>14</td>
<td>18</td>
</tr>
<tr>
<td>19</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>79</td>
<td>100</td>
<td>21</td>
<td>27</td>
</tr>
</tbody>
</table>
gain-in-complete, was recorded for design team $i = 9$ which had a 14% loss. The maximum for growth-in-complete, was recorded for design team $i = 15$ which grew by 100%. The minimum for growth-in-complete, was recorded for design team $i = 9$ which had a 14% loss.

We believe that these numbers provide remarkable empirical evidence for the usefulness of Armstrong relations for the acquisition of meaningful FDs in terms of completeness.

7.3. Proximity

The proximity ratios of the FD sets $\Sigma_1^i$ and $\Sigma_2^i$ relative to the target set $\Sigma^T$ are recorded in Table 9 and illustrated in Fig. 6.

Table 9

Proximity analysis.

<table>
<thead>
<tr>
<th>Team $i$</th>
<th>$\text{prox-ratio}_{1}^1(\Sigma_1^1)$ in percent</th>
<th>$\text{prox-ratio}_{2}^2(\Sigma_2^2)$ in percent</th>
<th>gain-in-prox, in percent</th>
<th>growth-in-prox, in percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>57</td>
<td>50</td>
<td>–7</td>
<td>–12</td>
</tr>
<tr>
<td>2</td>
<td>79</td>
<td>79</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>64</td>
<td>100</td>
<td>36</td>
<td>56</td>
</tr>
<tr>
<td>4</td>
<td>64</td>
<td>93</td>
<td>29</td>
<td>45</td>
</tr>
<tr>
<td>5</td>
<td>71</td>
<td>86</td>
<td>15</td>
<td>21</td>
</tr>
<tr>
<td>6</td>
<td>57</td>
<td>86</td>
<td>29</td>
<td>51</td>
</tr>
<tr>
<td>7</td>
<td>57</td>
<td>64</td>
<td>7</td>
<td>12</td>
</tr>
<tr>
<td>8</td>
<td>71</td>
<td>71</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>57</td>
<td>79</td>
<td>22</td>
<td>39</td>
</tr>
<tr>
<td>10</td>
<td>79</td>
<td>57</td>
<td>–22</td>
<td>–27</td>
</tr>
<tr>
<td>11</td>
<td>71</td>
<td>93</td>
<td>22</td>
<td>31</td>
</tr>
<tr>
<td>12</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>64</td>
<td>93</td>
<td>29</td>
<td>45</td>
</tr>
<tr>
<td>15</td>
<td>43</td>
<td>86</td>
<td>43</td>
<td>100</td>
</tr>
<tr>
<td>16</td>
<td>93</td>
<td>100</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>17</td>
<td>64</td>
<td>100</td>
<td>36</td>
<td>56</td>
</tr>
<tr>
<td>18</td>
<td>79</td>
<td>93</td>
<td>14</td>
<td>18</td>
</tr>
<tr>
<td>19</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>79</td>
<td>100</td>
<td>21</td>
<td>27</td>
</tr>
</tbody>
</table>
Table 9 shows that \( \text{gain-in-prox} \) in percent was 14.05, and \( \text{growth-in-prox} \) in percent was 20.1. According to our terminology, the use of Armstrong relations for the acquisition of meaningful FDs achieved a gain of 14% and a growth by 20% in terms of proximity. The results follow from the definition of soundness, completeness and proximity.

Another important observation is that apart from two exceptions \((i = 1 \text{ and } 10)\) we did not record any decrease in the proximity, i.e., \( \text{gain-in-prox} \geq 0 \) and \( \text{growth-in-prox} \geq 1 \) holds for all \( i \in \{1, \ldots, 20\} \setminus \{1, 10\} \) (90% of all design teams). Three out of these 18 design teams \((i = 12, 13 \text{ and } 19)\) already achieved 100% proximity in phase 1. Moreover, 12 design teams had an actual increase in proximity, i.e., \( \text{gain-in-prox} > 0 \) and \( \text{growth-in-prox} > 1 \) holds for 60% of all design teams. In fact, all of the 7 design teams that (i) did not incorrectly specify any meaningless FDs and (ii) did not already have 100% proximity after phase 1, achieved an actual increase. Three of these 7 design teams \((i = 16, 17 \text{ and } 20)\) achieved 100% proximity. Furthermore, one additional design team \((i = 3)\) also achieved 100% proximity. All together, we had 7 out of 20 design teams that specified a cover of the target set at the end of the experiment (35% of all design teams). Note that for the design teams that did not incorrectly specify any meaningless FDs \((1 \leq i \leq 20)\) completeness and proximity are the same.

The maximum for \( \text{gain-in-prox} \) was recorded for design team \( i = 15 \) which gained 43%. The minimum for \( \text{gain-in-prox} \) was recorded for design team \( i = 10 \) which had a 22% loss.

The maximum for \( \text{growth-in-prox} \) was recorded for design team \( i = 15 \) which gained 100%. The minimum for \( \text{growth-in-prox} \) was recorded for design team \( i = 10 \) which had a 27% loss.

We believe that these numbers provide remarkable empirical evidence for the usefulness of Armstrong relations for the acquisition of meaningful FDs in terms of proximity.

7.4. Minimality

The minimality ratios of the FD sets \( \Sigma_1^i \) and \( \Sigma_2^i \) are recorded in Table 10 and illustrated in Fig. 7.

Table 10 shows that \( \text{gain-in-min} \) in percent was 4.45, and that \( \text{growth-in-min} \) in percent was 7.04. According to our terminology, the use of Armstrong relations for the acquisition of meaningful FDs achieved a gain of 4.5% and a growth by 7% in terms of minimality.

The maximum for \( \text{gain-in-min} \) was recorded for design teams \( i = 2 \) and 13 which both gained 50%. The minimum for \( \text{gain-in-min} \) was recorded for design team \( i = 6 \) which had a 25% loss.

The maximum for \( \text{growth-in-min} \) was recorded for design teams \( i = 2 \) and 13 which grew by 100%. The minimum for \( \text{growth-in-min} \) was recorded for design team \( i = 6 \) which had a 25% loss.

Strictly speaking, we do not believe that Armstrong relations have a real impact on the minimality of FD sets. The non-negligible gain of 4.5% and growth by 7% may have resulted from the additional time that was available.
to the design teams. If at all, the analysis shows additional consistency of the data: (i) all design teams that specified only meaningful FDs \((10 \leq i \leq 20)\) did not have a decrease in minimality, (ii) the three design teams that already had 100% proximity after phase 1 \((i = 12, 13\) and 19\) had actual gains of 23%, 50% and 20% and growth by 40%, 100% and 25% in minimality, and (iii) the three design teams that had decreases in minimality \((i = 1, 5\) and 6\) were occupied with correctly eliminating meaningless FDs \((i = 5\) and 6\) or correctly acquiring meaningful FDs \((i = 1\) and 6\).

## 8. Qualitative data analysis

In this section, we present and analyze the different quality measures that the design teams achieved during our experiment. We are using the following abbreviations

### Table 10
Minimality analysis.

<table>
<thead>
<tr>
<th>Team i</th>
<th>(min\text{-}ratio_{\frac{\Sigma A_i}{\Sigma \frac{\Sigma A_i}{\Sigma A_i}}}) in percent</th>
<th>(min\text{-}ratio_{\frac{\Sigma B_i}{\Sigma \frac{\Sigma B_i}{\Sigma B_i}}}) in percent</th>
<th>gain-in-min, in percent</th>
<th>growth-in-min, in percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>80</td>
<td>-20</td>
<td>-20</td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>100</td>
<td>50</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td>80</td>
<td>-20</td>
<td>-20</td>
</tr>
<tr>
<td>6</td>
<td>100</td>
<td>75</td>
<td>-25</td>
<td>-25</td>
</tr>
<tr>
<td>7</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>57</td>
<td>57</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>56</td>
<td>57</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>57</td>
<td>80</td>
<td>23</td>
<td>40</td>
</tr>
<tr>
<td>13</td>
<td>50</td>
<td>100</td>
<td>50</td>
<td>100</td>
</tr>
<tr>
<td>14</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>75</td>
<td>80</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>17</td>
<td>75</td>
<td>80</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>18</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>80</td>
<td>100</td>
<td>20</td>
<td>25</td>
</tr>
<tr>
<td>20</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Fig. 7. Minimality before and after using Armstrong relations.

Some statistics on the gains/growths after using Armstrong relations:
- Gains: Minimum: -25%, Arithmetic Mean: 4%, Maximum: 50%
- Growths: Minimum: -25%, Geometric Mean: 7%, Maximum: 100%, Median: 0%
Armstrong relations support this quantitative analysis. We have also argued how the properties of correctly perceived as meaningful, and to (ii) incorrectly to (i) correctly identify meaningless FDs that are in-
tively speaking, in using Armstrong relations it is unlikely

8.1. Soundness
We have seen in the previous section that, quantita-
tively speaking, in using Armstrong relations it is unlikely

to (i) correctly identify meaningless FDs that are in-

correctly perceived as meaningful, and to (ii) incorrectly

Correctly discovered by more than one design team was

Room

Time

L_Name

FDs are incorrectly removed. We will now provide some qualitative evidence for these observations.

The first three left-hand columns of Table 13 show for the design teams \( i \) which attribute sets \( X \) include attributes in their closures with respect to the target set \( \Sigma^i \) that are not in their closures with respect to \( \Sigma^1 \) and \( \Sigma^2 \), respectively. We can observe easily (i) which of the attribute subsets that occur in \( \text{incomplete}_{\Sigma^i}(\Sigma^1) \) do not occur in \( \text{incomplete}_{\Sigma^i}(\Sigma^2) \), and (ii) which of the attribute subsets that do not occur in \( \text{incomplete}_{\Sigma^i}(\Sigma^1) \) do occur in \( \text{incomplete}_{\Sigma^i}(\Sigma^2) \).

Therefore, Table 13 provides us with specific exceptions to the average rules that we observed above. For example, we have \( \text{incomplete}_{\Sigma^2}(\Sigma^1) = \text{incomplete}_{\Sigma^2}(\Sigma^2) \) = (TR), or \( C \in \text{incomplete}_{\Sigma^2}(\Sigma^1) \) – \( \text{incomplete}_{\Sigma^2}(\Sigma^2) \).

Moreover, column three indicates which meaningful FDs were frequently not perceived as meaningful, even after inspection of the Armstrong relations. These include C_ID – L_Name,

L_Name, Time → C_ID,

L_Name, Time → Room, and

Time, Room → C_ID.

Table 12 shows which additional meaningful FDs were correctly discovered by the design teams after inspecting Armstrong relations. Noticeably, there were several teams that discovered the FDs L_Name, Time → C_ID or L_Name, Time → Room. Firstly, these FDs were too complex to be recognized as meaningful by the teams without an Armstrong relation. Secondly, an inspection of the Armstrong relation made the teams realize that these are potentially meaningful FDs. In fact, the FD L_Name, Time → Room may be specified without consultation of a domain expert (no lecturer can be in two different rooms at the same time). For the FD L_Name, Time → C_ID, some of the teams decided to ask the domain experts whether it is possible that different courses can be co-taught by the same lecturer at the same time. Another meaningful FD that was frequently discovered after inspecting an Armstrong relation was C_ID → L_Name. It is likely that design teams for whom this was the case did not consider the FD prior to the inspection of an Armstrong relation. It seems that the

Table 11
Qualitative soundness analysis.

<table>
<thead>
<tr>
<th>( i )</th>
<th>( \text{unsound}_{\Sigma^i}(\Sigma^1) )</th>
<th>( \text{unsound}_{\Sigma^i}(\Sigma^2) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CLR</td>
<td>R, LR, CR, CLR</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>CR, CLR</td>
</tr>
<tr>
<td>3</td>
<td>L, LR</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>LR</td>
</tr>
<tr>
<td>5</td>
<td>CR, CLR</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>LR</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>C, CL, CR, LR, CLR</td>
<td>C, CL, CR, LR, CLR</td>
</tr>
<tr>
<td>8</td>
<td>LR, CLR</td>
<td>C, CL, CR, CLR</td>
</tr>
<tr>
<td>9</td>
<td>C, L, CL, CR, LR, CLR</td>
<td>L</td>
</tr>
<tr>
<td>10</td>
<td>L, LR</td>
<td>L, R, LR, CR, CLR</td>
</tr>
</tbody>
</table>

for sake of clearer presentation: C for C_ID, L for L_Name, T for Time, and R for Room.

8.1. Soundness

We have seen in the previous section that, quantita-
tively speaking, in using Armstrong relations it is unlikely

to (i) correctly identify meaningless FDs that are in-

correctly perceived as meaningful, and to (ii) incorrectly

Correctly discovered by more than one design team was

Room

Time

L_Name

FDs are incorrectly removed. We will now provide some qualitative evidence for these observations.

The first three left-hand columns of Table 13 show for the design teams \( i \) which attribute sets \( X \) include attributes in their closures with respect to the target set \( \Sigma^i \) that are not in their closures with respect to \( \Sigma^1 \) and \( \Sigma^2 \), respectively. We can observe easily (i) which of the attribute subsets that occur in \( \text{incomplete}_{\Sigma^i}(\Sigma^1) \) do not occur in \( \text{incomplete}_{\Sigma^i}(\Sigma^2) \), and (ii) which of the attribute subsets that do not occur in \( \text{incomplete}_{\Sigma^i}(\Sigma^1) \) do occur in \( \text{incomplete}_{\Sigma^i}(\Sigma^2) \).

Therefore, Table 13 provides us with specific exceptions to the average rules that we observed above. For example, we have \( \text{incomplete}_{\Sigma^2}(\Sigma^1) = \text{incomplete}_{\Sigma^2}(\Sigma^2) \) = (TR), or \( C \in \text{incomplete}_{\Sigma^2}(\Sigma^1) \) – \( \text{incomplete}_{\Sigma^2}(\Sigma^2) \).

Moreover, column three indicates which meaningful FDs were frequently not perceived as meaningful, even after inspection of the Armstrong relations. These include C_ID → L_Name,

L_Name, Time → C_ID,

L_Name, Time → Room, and

Time, Room → C_ID.

Table 12 shows which additional meaningful FDs were correctly discovered by the design teams after inspecting Armstrong relations. Noticeably, there were several teams that discovered the FDs L_Name, Time → C_ID or L_Name, Time → Room. Firstly, these FDs were too complex to be recognized as meaningful by the teams without an Armstrong relation. Secondly, an inspection of the Armstrong relation made the teams realize that these are potentially meaningful FDs. In fact, the FD L_Name, Time → Room may be specified without consultation of a domain expert (no lecturer can be in two different rooms at the same time). For the FD L_Name, Time → C_ID, some of the teams decided to ask the domain experts whether it is possible that different courses can be co-taught by the same lecturer at the same time. Another meaningful FD that was frequently discovered after inspecting an Armstrong relation was C_ID → L_Name. It is likely that design teams for whom this was the case did not consider the FD prior to the inspection of an Armstrong relation. It seems that the

Table 12
FD analysis—soundness.

<table>
<thead>
<tr>
<th>( i )</th>
<th>Meaningless FD added</th>
<th>Meaningless FD removed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R → T</td>
<td>–</td>
</tr>
<tr>
<td>2</td>
<td>CR → T</td>
<td>–</td>
</tr>
<tr>
<td>3</td>
<td>–</td>
<td>L → C</td>
</tr>
<tr>
<td>4</td>
<td>LR → C</td>
<td>–</td>
</tr>
<tr>
<td>5</td>
<td>–</td>
<td>CR → T</td>
</tr>
<tr>
<td>6</td>
<td>–</td>
<td>LR → C</td>
</tr>
<tr>
<td>7</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>8</td>
<td>C → T, C → R</td>
<td>–</td>
</tr>
<tr>
<td>9</td>
<td>–</td>
<td>C → T, C → R, LR → C</td>
</tr>
<tr>
<td>10</td>
<td>R → T</td>
<td>–</td>
</tr>
</tbody>
</table>
the FD team i was inspecting the Armstrong relations. In particular, design meaningless FDs fact, this was a result of the team’s removal of the S were incorrectly removed from the FD sets that this could indeed be a meaningful FD. Armstrong relation pointed these teams to the possibility that this could indeed be a meaningful FD.

Table 14 also shows that only two meaningful FDs were incorrectly removed from the various FD sets \( \Sigma_1 \), that is, where \( \sigma \) is implied by \( \Sigma - \{ \sigma \} \).

Table 14
<table>
<thead>
<tr>
<th>i</th>
<th>Meaningful FD added</th>
<th>Meaningful FD removed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LT ( \rightarrow ) R</td>
<td>–</td>
</tr>
<tr>
<td>2</td>
<td>CT ( \rightarrow ) R</td>
<td>–</td>
</tr>
<tr>
<td>3</td>
<td>C ( \rightarrow ) L</td>
<td>–</td>
</tr>
<tr>
<td>4</td>
<td>LT ( \rightarrow ) R, TR ( \rightarrow ) L</td>
<td>–</td>
</tr>
<tr>
<td>5</td>
<td>TR ( \rightarrow ) C</td>
<td>CR ( \rightarrow ) L</td>
</tr>
<tr>
<td>6</td>
<td>LT ( \rightarrow ) R</td>
<td>–</td>
</tr>
<tr>
<td>7</td>
<td>LT ( \rightarrow ) C</td>
<td>–</td>
</tr>
<tr>
<td>8</td>
<td>C ( \rightarrow ) L</td>
<td>–</td>
</tr>
<tr>
<td>9</td>
<td>–</td>
<td>C ( \rightarrow ) L</td>
</tr>
<tr>
<td>10</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>11</td>
<td>CT ( \rightarrow ) R</td>
<td>–</td>
</tr>
<tr>
<td>12</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>13</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>14</td>
<td>C ( \rightarrow ) L, CT ( \rightarrow ) R</td>
<td>–</td>
</tr>
<tr>
<td>15</td>
<td>CT ( \rightarrow ) L, TR ( \rightarrow ) L, LT ( \rightarrow ) C</td>
<td>–</td>
</tr>
<tr>
<td>16</td>
<td>TR ( \rightarrow ) C</td>
<td>–</td>
</tr>
<tr>
<td>17</td>
<td>C ( \rightarrow ) L</td>
<td>–</td>
</tr>
<tr>
<td>18</td>
<td>LT ( \rightarrow ) C</td>
<td>–</td>
</tr>
<tr>
<td>19</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>20</td>
<td>CT ( \rightarrow ) R, LT ( \rightarrow ) C</td>
<td>–</td>
</tr>
</tbody>
</table>

Armstrong relation pointed these teams to the possibility that this could indeed be a meaningful FD.

Table 14 also shows that only two meaningful FDs were incorrectly removed from the FD sets \( \Sigma_1 \) after inspecting the Armstrong relations. In particular, design team i = 9 implicitly removed the FD C ID \( \rightarrow \) L_Name. In fact, this was a result of the team’s removal of the meaningless FDs C ID \( \rightarrow \) Room and C ID \( \rightarrow \) Time after which the FD C ID \( \rightarrow \) L_Name was no longer implied by \( \Sigma_2 \).

8.3. Proximity and minimality

For proximity, we simply notice that dist(\( \Sigma, \Sigma_2 \)) = unsound_\( \Sigma_2 \)(\( \Sigma \)) \cup incomplete_\( \Sigma \)(\( \Sigma \)) holds. Consequently, the analysis follows from the analysis of the soundness and completeness.

A qualitative analysis of the minimality would not add any new observations. It would simply mean to list the FDs \( \sigma \) that are redundant in the various FD sets \( \Sigma \), that is, where \( \sigma \) is implied by \( \Sigma - \{ \sigma \} \).

9. Conclusions

We have conducted a first empirical investigation about the usefulness of Armstrong relations for the acquisition of meaningful functional dependencies. Specifically, we introduced the three measures of soundness, completeness, and proximity to study the additional insights that one can obtain from the inspection of Armstrong relations. The first main result indicates that Armstrong relations are not useful in terms of soundness, i.e., in using Armstrong relations it is not more likely to recognize meaningless FDs which are incorrectly perceived as meaningful. The second main result indicates that Armstrong relations are indeed useful in terms of completeness, i.e., in using an Armstrong relation it is more likely to recognize meaningful FDs that are incorrectly perceived as meaningless. The results are intuitive as it seems unlikely to recognize the satisfaction of meaningless FDs, but it seems relatively likely to recognize the violation of meaningful FDs by the given Armstrong relation. We believe that these results provide new insight on the usefulness of Armstrong relations. We hope that (i) database designers will be motivated by our findings to utilize Armstrong databases effectively during the requirements elicitation process, (ii) database researchers will further investigate the properties of Armstrong databases, and (iii) database instructors will be able to improve the efficiency of their teaching by spending less time on marking.
10. Future directions

There are various avenues that should be explored in future work.

One avenue should address the limitations that we mentioned previously. In particular, empirical evidence should be collected in a range of different application domains, from real database designers, under more realistic time constraints, in the absence or partial availability of domain experts and with differences in the experts’ opinions. Our study can provide exact means to conduct such experiments.

Another direction is the study of different classes of data dependencies that enjoy Armstrong relations. Most interestingly, it might be worthwhile to investigate keys only, keys and foreign keys, multivalued dependencies [86,87] and standard functional dependencies together with inclusion dependencies. This will also make it necessary to explore the generalization of our measures.

Furthermore, it would be interesting to explore the impact of the sizes of Armstrong relations on their usefulness. In the literature it is stated that small Armstrong relations are more beneficial. The reason is that design teams are better able to comprehend small examples. It would be interesting to provide empirical evidence for this as well. For example, is it true that Armstrong relations that only realize the maximal sets induced by an FD set allow us to recognize a higher number of meaningful FDs than Armstrong relations that realize all closed sets induced by the FD set [9,12]?

Moreover, we would like to point out that it is worthwhile to develop a process model for utilizing Armstrong relations effectively in the requirements analysis phase of a target database. For example, our main results indicate to utilize Armstrong relations as early as possible in order to prevent design teams from incorrectly perceiving any meaningless FDs as meaningful.

The computation of our measures for the quality of a submitted FD set relative to a target FD set should be implemented as an automated assessment and feedback tool in database courses.

Finally, our investigations should be extended to the concept of informative Armstrong databases [14]. For example, one may ask in what precise sense and to which degree informative Armstrong databases can be useful for the acquisition of existing or new database dependencies.

Acknowledgments

This research is supported by the Marsden fund council from Government funding, administered by the Royal Society of New Zealand.

We would like to thank Tiong Goh, Sven Hartmann, Pavle Mogin and Dion Peszynski for their kind assistance with the data collection for this project as well as their suggestions to improve the clarity of our presentation. We are also grateful to Dennis Shasha for his suggestions and comments that resulted in an improvement of the motivation and presentation of our results.

References


A. Silva, M. Melkankoff, A method for helping discover the dependencies of a relation, in: Advances in Data Base Theory, 1979, pp. 115–133.


