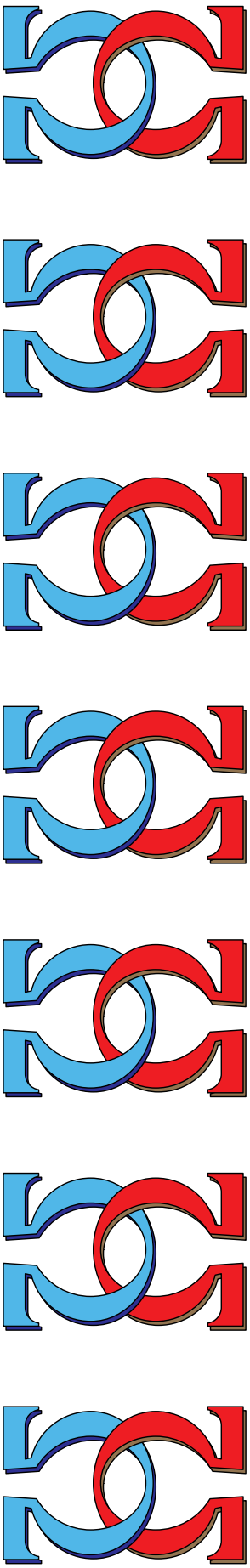

CDMTCS
Research
Report
Series

Possible and Certain SQL Keys

Henning Köhler
Massey University
Palmerston North, New Zealand

Uwe Leck
University of Wisconsin-Superior
Superior, WI, U.S.A.

Sebastian Link
University of Auckland,
Auckland, New Zealand

CDMTCS-452
December 2013

Centre for Discrete Mathematics and
Theoretical Computer Science

Possible and Certain SQL Keys

Henning Köhler
Massey University
Palmerston North, New Zealand
h.koehler@massey.ac.nz

Uwe Leck
University of Wisconsin-Superior
Superior, WI, U.S.A.
uleck@uwsuper.edu

Sebastian Link
The University of Auckland, Private Bag 92019
Auckland, New Zealand
s.link@auckland.ac.nz

December 17, 2013

Abstract

In standard SQL database management systems primary key columns are NOT NULL by default. While NULL columns may be included in unique constraints, such constraints only ensure uniqueness for tuples which do not feature any null marker occurrences in the columns involved, and do not fulfil the same function as primary keys. In this work we investigate the notions of possible and certain keys, which are intuitive and differ only in their treatment of null markers. It turns out that possible keys capture the unique constraint of SQL, while certain keys extend primary keys to include NULL columns, and can be used for similar purposes. In addition to basic characterization, axiomatization, and simple discovery approaches for possible and certain keys, we investigate the existence and construction of Armstrong tables, extremal set problems, and describe an indexing scheme for enforcing certain keys. Our experiments show that certain keys with NULLs do occur in real-world databases, and that related computational problems can be solved efficiently. Certain keys are semantically well-founded, achieve the goal of Codd's entity integrity rule and offer more flexibility for data entry than primary keys.

Keywords: Armstrong database, Axiomatization, Complexity, Cover, Discovery, Extremal combinatorics, Implication problem, Index, Key, Null marker, Possible world, Update

1 Introduction

Entity integrity is one of Codd's integrity rules which states that every table must have a primary key and that the columns which form the primary key should be unique and

not null [10]. The goal of entity integrity is to ensure that every tuple in the table can be identified efficiently. In SQL, entity integrity is enforced by adding a primary key clause to a schema definition. The system enforces entity integrity by not allowing operations, that is inserts and updates, that produce an invalid primary key. Operations that create a duplicate primary key or one containing nulls are rejected.

Consider snapshot I of the RFAM (RNA families) data set in Table 1, available at `http://rfam.sanger.ac.uk/`. The data set violates entity integrity as every potential primary key over the schema is violated by I. In particular, column *journal* carries the null marker $\perp$. Nevertheless, every tuple in I can be uniquely identified by a combination of column pairs, that is, by (title, journal) or also by (author, journal). This property cannot be enforced by SQL's `unique` constraint, as these cannot always uniquely identify tuples in which null markers occur in the columns involved. We conclude that the syntactic requirements of primary keys are sufficient to meet the goal of entity integrity, but not necessary. Indeed, primary keys prohibit the entry of some data without good reason. This should not be underestimated as the inability to enter important data into the database may force organizations to abandon key validation altogether, exposing their future database to less data quality, inefficiencies in data processing, waste of resources, and poor data-driven decision making. In other words, finding a notion of keys that is sufficient and necessary to meet the goal of entity integrity would bring forward a new database technology that is more useful in practice.

title	*author*	*journal*
The uRNA database	Zwieb C	Nucleic Acids 1997
The uRNA database	Zwieb C	Nucleic Acids 1996
Genome wide detect.	Ragh. R	$\perp$

Table 1: A snippet of the RFAM data set

These observations have motivated us to investigate keys over SQL tables from a well-founded semantic point of view. For this purpose, we adopt Codd's well-accepted interpretation of null marker occurrences as "value exists, but unknown", which we denote by $\perp$. This interpretation leads naturally to a possible world semantics, in which a possible world results from a table by replacing independently each occurrence of $\perp$ by a value from the corresponding domain. A possible world is therefore a multiset of total tuples. As usual, a possible world w satisfies a key X if and only if there are no two different tuples in w that have matching values on all the attributes in X. Hence, a key can only be satisfied by a possible world if it is a set of tuples. This approach naturally suggests two semantics of keys over SQL tables. A *possible key* $p\langle X\rangle$ is satisfied by an SQL table I if and only if there is a possible world of I in which the key X is satisfied. A *certain key* $c\langle X\rangle$ is satisfied by an SQL table I if and only if the key X is satisfied in every possible world of I. In particular, the semantics of certain keys does not prevent the entry of incomplete tuples that can still be identified uniquely, independently of which values null marker occurrences represent. For examples, the snapshot I in Table 1 satisfies the certain keys $c\langle title, journal\rangle$ and $c\langle author, journal\rangle$. In fact, I forms a subset of the RFAM data set that satisfies the same possible and certain keys and `NOT`

NULL columns as the entire data set. For space reasons we have omitted two columns here and have abbreviated some values.

Our observations provide strong motivation to investigate possible and certain keys in detail. In particular, it is interesting to find out whether certain keys can meet the goal of entity integrity to identify tuples *efficiently*. The contributions of our research can be summarized as follows:

- We propose a possible world semantics for keys over SQL tables. Possible keys hold in some possible world, while certain keys hold in all possible worlds. Hence, certain keys identify tuples without having to declare all attributes NOT NULL. While primary keys provide a sufficient condition to identify tuples, the condition is not necessary. Certain keys provide a sufficient and necessary condition, thereby not prohibiting the entry of data that can be uniquely identified.

- We establish simple syntactic characterizations to validate the satisfaction of possible and certain keys. In fact, possible keys provide a semantics for SQL's unique constraint. That is, unique(X) is satisfied by an SQL table if and only if it satisfies $p\langle X\rangle$.

- We characterize the implication problem for the combined class of possible and certain keys and NOT NULL constraints axiomatically and by an algorithm that works in linear time in the input. This shows that possible and certain keys can be reasoned about efficiently. As an important application, we can efficiently compute *the* minimal cover of our constraints in order to reduce the amount of integrity maintenance to a minimal level necessary.

- We address the data-driven discovery of possible and certain keys. Exploiting hypergraph transversals, we establish a compact algorithm to compute a cover for the set of possible and certain keys that hold on a given table. The discovery algorithm allows us to find possible and certain keys that hold on publicly available biological data sets. In particular, several of the certain keys permit null marker occurrences, which makes them different from primary keys. Hence, certain keys do occur in practice, providing strong motivation to do further research on them and exploit their useful features in future database technology.

- We then investigate structural and computational aspects of Armstrong tables for our combined class of constraints. In particular, we characterize when Armstrong tables exist and how to compute them in these cases. We also provide circumstantial evidence that deciding the existence of Armstrong tables is likely to be intractable in the general case. Armstrong tables provide a tool for database designers to communicate effectively with domain experts in order to acquire a more complete set of semantically meaningful integrity constraints. It is well-known that this results in better database designs, better data quality, more efficient data processing, exchange and integration, resource savings and better decision-making [36].

- For a database designer it is a natural question to ask how large non-redundant families of integrity constraints can potentially be. Answers to this question pro-

vide the designer with upper bounds on how complex integrity maintenance could become. This may result in the requirement to restrict the size of keys. One may then ask how large non-redundant families of restricted keys can become. Using extremal set theory we identify the non-redundant families of possible and certain keys that attain maximum cardinality, even when we limit the keys to those that respect an upper bound on the number of attributes.

- We propose an indexing scheme for certain keys. Our scheme improves the enforcement of certain keys on inserts by several orders of magnitude. It works only marginally slower than the enforcement of primary keys, provided that the certain keys have only a small number of columns in which null markers can occur. Exploiting our data-driven discovery algorithm from before, we have found only certain keys in which at most two columns can feature null markers.

- Besides the discovery of possible and certain keys in real-life data sets we conducted several other experiments. These confirm our intuition that the computational problems above can be solved efficiently in practice. For example, we applied our construction of Armstrong tables to the possible and certain keys we had previously discovered, resulting in tables containing only seven rows on average, which makes them particularly useful as a communication tool. Furthermore, their computation only took a few milliseconds in each of the 130 cases. For only 85 out of 1 million randomly generated schemata and sets of keys, Armstrong tables did not exist, otherwise Armstrong tables were computed in a few milliseconds. Although the computation of Armstrong tables may take exponential time in the worst case, such cases need to be constructed carefully and occur at best sparingly in practice or by chance. Finally, experiments with our scheme showed that i) certain keys in practice are likely to not have more than two columns in which null markers occur, and ii) such certain keys can be enforced almost as efficiently as primary keys. Our findings provide strong evidence that certain keys achieve the goal of Codd's principle of entity integrity. The choice between primary and certain keys should be based on the requirements for data entry.

Organization. The remainder of the paper is organized as follows. Section 2 discusses related work further motivating our research. Possible and certain keys are introduced in Section 3 where we also establish their syntactic characterization, axiomatic and algorithmic solutions to their implication problem, the computation of their minimal cover, and their discovery from given tables. Structural and computational aspects of Armstrong tables are investigated in Section 4. Extremal problems of possible and certain keys are studied in Section 5. An efficient indexing scheme for the enforcement of certain keys is established in Section 6, and results of our experiments are presented in Section 7. We conclude and comment on future work in Section 8. Proofs and further material have been moved to the appendix.

2 Related Work

Integrity constraints directly address the semantics of data and thereby form one of the cornerstones of database theory and practice [1]. Besides domain and referential integrity, entity integrity is one of the three inherent integrity rules proposed by Codd [10]. These three basic types of integrity constraints are special as they are the only ones, amongst around 100 different classes of constraints [1], that enjoy built-in support by SQL database management systems. In particular, entity integrity is enforced directly by the primary key mechanism in SQL [38]. In the relational model, keys have been extended to more expressive classes of data dependencies, see [1, 19] for some excellent surveys. Core problems that are studied include axiomatic [3] and algorithmic [13] characterizations of the associated implication problem, Armstrong databases [5, 17, 22], data dependency discovery [36, 41], and extremal problems [28]. Important applications include database schema design [16, 36], query optimization [12], transaction processing [2], view maintenance [29], data exchange [18, 37], data integration [8], data cleaning [20], and data security [6], to name a few.

One of the most important extensions of Codd's basic relational model [10] is incomplete information. This is mainly due to the high demand for the correct handling of such information in real-world applications. While there are many approaches to incomplete information the focus of this paper is on null markers in SQL. In the literature many kinds of null markers have been proposed. The two most prolific interpretations are "value unknown at present" [9] and "no information" [45], on which we will focus in this section.

We first consider the interpretation "value unknown at present". In this context, Levene and Loizou introduced the notions of strong and weak functional dependency (FD) [33], using a possible world semantics. Our definitions of possible/certain keys take the same approach, and strong/weak FDs and possible/certain keys are closely related. However, neither are certain keys special cases of strong FDs, nor are possible keys special cases of weak FDs. The focus in [33] is on axiomatization and implication of strong/weak FDs. For keys these problems turn out to be simpler, allowing us to focus on other issues, in particular those associated with Armstrong tables. While Armstrong tables are claimed to exist for any set of weak/strong FDs [33], the proof contains a technical error. In Section 4 we give an example for a set of possible and certain keys and `NOT NULL` constraints for which no Armstrong table exists. Since possible/certain keys are not a special case of weak/strong FDs and `NOT NULL` constraints are not considered in [33] the example does not directly contradict the result in [33]. However, our example requires only minor modification to show that not every set of weak and strong FDs can be represented in form of an Armstrong table. Such a modified example is given in Section B of the appendix. Therefore, Armstrong tables require renewed investigation for weak and strong FDs. The combined class of weak FDs and `NOT NULL` constraints is studied in [21] discussing implication, discovery and construction of Armstrong tables. As mentioned before, already possible keys by themselves are not a special case of weak FDs.

We turn to the interpretation "no information", for which FDs and multivalued dependencies (MVDs) have been investigated [4, 25, 35, 45]. The approach is syntactic and

different from the possible world semantics. However, some classes of constraints enjoy the same axiomatization under both interpretations of null markers. Keys in the context of the "no information" interpretation were studied in [31, 32], where implication, discovery and construction of Armstrong tables are considered. These capture exactly the `unique` constraint of SQL. One of our results in the present paper therefore also shows that possible keys do not just capture SQL's `unique` constraint but also the keys from [31, 32]. The class of FDs and MVDs under the "no information" interpretation coincides with the class of weak FDs and weak MVDs under a possible world semantics [25]. The existence and computation of Armstrong tables was investigated in [22] for the combined class of keys, FDs, and `NOT NULL` constraints under the "no information" interpretation.

As final related work we want to discuss two different approaches. The principle of entity integrity has been first challenged by Thalheim [43] and later by Levene and Loizou [34], both following an approach different from ours. As an alternative, they propose the notion of a key set. A relation satisfies a key set if, for each pair of distinct tuples, there is some key in the key set on which the two tuples are total and distinct. A certain key is equivalent to a key set consisting of all the singleton subsets of the key attributes, e.g., $c\langle A, B, C\rangle$ corresponds to the key set $\{\{A\}, \{B\}, \{C\}\}$. However, our work is different in that we study the interaction with possible keys and `NOT NULL` attributes, establish a possible world semantics, and study different problems. The implication problem for the sole class of primary keys is examined in [23]. As the key columns of primary keys are `NOT NULL`, the class of primary keys behaves differently from both possible and certain keys, see Section 3.3.

We emphasize that our findings may also be important for other data models such as XML [7, 11, 24] and RDF [39] where incomplete information is inherent, probabilistic databases [27] where keys can be expected to hold with some probability other than 1, and also description logics [44].

Summary. Certain keys appear to be the most natural approach to address the efficient identification of tuples in SQL tables. It is therefore surprising that they have not been considered in previous work. In this paper, we investigate the combined class of possible and certain keys under `NOT NULL` constraints. The combination of these constraints is particularly relevant to SQL, as possible keys correspond to SQL's `unique` constraint. The presence of certain keys also means that the problems studied here are substantially different from those investigated elsewhere.

3 Possible and Certain Keys

In this section we first give some preliminary definitions before we introduce the notions of possible and certain keys, based on a possible world semantics. Subsequently, we characterize these notions syntactically, from which we derive both a simple axiomatic and a linear time algorithmic characterization of the associated implication problem. We show that possible and certain keys enjoy a unique minimal representation. Finally, we exploit hypergraph transversals to discover possible and certain keys from a given table.

3.1 Preliminaries

We begin with basic terminology. Let $\mathfrak{A} = \{A_1, A_2, \ldots\}$ be a (countably) infinite set of distinct symbols, called *attributes*. Attributes represent column names of tables. A *table schema* is a finite non-empty subset T of $\mathfrak{A}$. Each attribute A of a table schema T is associated with an infinite domain $dom(A)$ which represents the possible values that can occur in column A. In order to encompass incomplete information the domain of each attribute contains the null marker, denoted by $\perp$. The interpretation of $\perp$ is to mean "value unknown at present". We stress that the null marker is not a domain value. In fact, it is a purely syntactic convenience that we include the null marker in the domain of each attribute as a distinguished element.

For attribute sets X and Y we may write XY for their set union $X \cup Y$. If $X = \{A_1, \ldots, A_m\}$, then we may write $A_1 \cdots A_m$ for X. In particular, we may write A to represent the singleton $\{A\}$. A *tuple* over T is a function $t : T \to \bigcup_{A \in T} dom(A)$ with $t(A) \in dom(A)$ for all $A \in X$. For $X \subseteq T$ let $t[X]$ denote the restriction of the tuple t over T to X. We say that a tuple t is *X-total* if $t[A] \neq \perp$ for all $A \in X$. A tuple t over T is said to be a *total tuple* if it is T-total. A *table* I over T is a finite multiset of tuples over T. A table I over T is a *total table* if every tuple $t \in I$ is total.

Definition 1 (Strong/Weak Similarity)
Let t, t' be tuples over T. We define weak/strong similarity *of t, t' on $X \subseteq T$ as follows:*

$$\begin{array}{rcl} t[X] \sim_w t'[X] & :\Leftrightarrow & \forall A \in X. \\ & & (t[A] = t'[A] \vee t[A] = \perp \vee t'[A] = \perp) \\ t[X] \sim_s t'[X] & :\Leftrightarrow & \forall A \in X. \\ & & (t[A] = t'[A] \neq \perp) \end{array}$$

Weak and strong similarity become identical for tuples that are X-total. In such "classical" cases we denote similarity by $t[X] \sim t'[X]$. We will use the phrase t, t' agree *interchangeably for t, t'* are similar*.*

A *null-free subschema* (NFS) over the table schema T is an expression T_S where $T_S \subseteq T$. The NFS T_S over T is satisfied by a table I over T if and only if I is T_S-total. SQL allows the specification of attributes as `NOT NULL`, so the set of attributes declared `NOT NULL` forms an NFS over the underlying table schema. For convenience we sometimes refer to the pair (T, T_S) as table schema.

We say that $X \subseteq T$ is a *key* for the total table I over T, denoted by $I \vdash X$, if and only if there are no two different tuples $t, t' \in I$ that agree on X. We will now define two different notions of keys over general tables, using possible world semantics.

Definition 2 (Possible/Certain Key)
Given a table I on T, a possible world *of I is obtained by independently replacing every occurrence of $\perp$ in I with a domain value. We say that $X \subseteq T$ is a* possible/certain key *for I, denoted by $p\langle X \rangle$ and $c\langle X \rangle$ respectively, if the following hold.*

$$I \vdash p\langle X \rangle :\Leftrightarrow X \text{ is a key for some possible world of } I$$
$$I \vdash c\langle X \rangle :\Leftrightarrow X \text{ is a key for every possible world of } I$$

Example 1 *For $T = \{title, author, journal\}$ let I denote Table 1:*

title	*author*	*journal*
The uRNA database	*Zwieb C*	*Nucleic Acids 1997*
The uRNA database	*Zwieb C*	*Nucleic Acids 1996*
Genome wide detect.	*Ragh. R*	$\perp$

Then $I \vdash p\langle journal\rangle$ as $\perp$ can be replaced by a domain value that is different from the two journals listed in I. Furthermore, $I \vdash c\langle title, journal\rangle$ as the first two rows can be distinguished on journal, and the last row can be distinguished on title, independently of the replacement for $\perp$. Finally, $I \vdash c\langle author, journal\rangle$ holds for similar reasons: the first two rows have different values on journal, and the last row has a unique value on author, independently of the replacement for $\perp$.

For a set Σ of constraints over table schema T we say that a table I over T *satisfies* Σ if I satisfies every $\sigma \in \Sigma$. If for some $\sigma \in \Sigma$ the table I does not satisfy σ we say that I *violates* σ (and violates Σ). A table I over (T, T_S) is a table I over T that satisfies T_S. A table I over (T, T_S, Σ) is a table I over (T, T_S) that satisfies Σ. When discussing possible and certain keys, the notions of strong and weak anti-keys will prove as useful as the notion of an anti-key has proven useful in discussing keys.

Definition 3 (Strong/Weak Anti-Key)
Let I be a table over (T, T_S) and $X \subseteq T$. We say that X is a strong/weak anti-key *for I, denoted by $\neg_p\langle X\rangle$ and $\neg_c\langle X\rangle$ respectively, if $p\langle X\rangle$ and $c\langle X\rangle$, respectively, do* not *hold on I. Here we also say that $\neg_p\langle X\rangle$ and/or $\neg_c\langle X\rangle$* hold *on I. We write $\neg\langle X\rangle$ to denote an anti-key which may be either strong or weak. A set Σ of constraints over (T, T_S)* permits *a set Π of strong and weak anti-keys if there exists a table I over (T, T_S, Σ) such that every anti-key in Π holds on I.*

Example 2 *For $T = \{title, author, journal\}$ let I denote the table over T from Table 1. Then $I \vdash \neg_p\langle title, author\rangle$ as the first two rows will agree on title and author in every possible world. Furthermore, $I \vdash \neg_c\langle journal\rangle$ as $\perp$ could be replaced by either of the two journals listed in I, resulting in possible worlds that violate the key $\{journal\}$.*

Note that permitting every single anti-key in a set is not the same as permitting the set of anti-keys as a whole. Checking whether Σ permits a single anti-key can be done easily using Theorem 2, while Theorem 9 shows that deciding whether Σ permits a set of anti-keys is NP-complete.

3.2 Syntactic Characterization

Possible and certain keys can be characterized syntactically using strong and weak similarity. For this we have made the assumptions that domains are infinite and tables finite.

Theorem 1 *$X \subseteq T$ is a possible (certain) key for I iff no distinct tuples in I are strongly (weakly) similar on X.*

Example 3 *For $T = \{title, author, journal\}$ let I denote Table 1 over T. Then $I \vdash p\langle journal\rangle$ as no two different tuples in I strongly agree on journal. Furthermore, $I \vdash c\langle title, journal\rangle$ as no two different tuples in I weakly agree on title and journal. Finally, $I \vdash c\langle author, journal\rangle$ as no two different tuples in I weakly agree on author and journal.*

3.3 Implication

In the following let (T, T_S) denote the schema under consideration. For a set $\Sigma \cup \{\varphi\}$ of constraints over (T, T_s) we say that Σ *implies* φ, denoted by $\Sigma \models \varphi$, if and only if every table over (T, T_S) that satisfies Σ also satisfies φ.

Theorem 2 *Let Σ be a set of possible and certain keys.*

i) Σ implies $c\langle X\rangle$ iff $c\langle Y\rangle \in \Sigma$ for some $Y \subseteq X$ or $p\langle Z\rangle \in \Sigma$ for some $Z \subseteq X \cap T_S$.

ii) Σ implies $p\langle X\rangle$ iff $c\langle Y\rangle \in \Sigma$ or $p\langle Y\rangle \in \Sigma$ for some $Y \subseteq X$.

Example 4 *Let $T = \{title, author, journal\}$ be our table schema, $T_S = \{title, author\}$ our NFS, and let Σ consist of $p\langle journal\rangle$, $c\langle title, journal\rangle$ and $c\langle author, journal\rangle$. Theorem 2 shows that Σ implies $c\langle title, author, journal\rangle$ and $p\langle title, journal\rangle$, but neither $c\langle journal\rangle$ nor $p\langle title, author\rangle$. This is independently confirmed by Table 1, which is an Armstrong table for (T, T_S, Σ). Indeed, Table 1 satisfies $c\langle title, author, journal\rangle$ and $p\langle title, journal\rangle$, but it violates $c\langle journal\rangle$ and $p\langle title, author\rangle$.*

Theorem 3 *Let Π be a set of strong and weak anti-keys.*

i) Π implies $\neg_p\langle X\rangle$ iff $\neg_p\langle Y\rangle \in \Pi$ for some $Y \supseteq X$, or $\neg_c\langle Z\rangle \in \Pi$ for some Z with $X \subseteq Z \cap T_S$.

ii) Π implies $\neg_c\langle X\rangle$ iff $\neg_c\langle Y\rangle \in \Pi$ or $\neg_p\langle Y\rangle \in \Pi$ for some $Y \supseteq X$.

Example 5 *Let $T = \{title, author, journal\}$ be our table schema, $T_S = \{title, author\}$ our NFS, and let Π consist of $\neg_p\langle title, author\rangle$ and $\neg_c\langle journal\rangle$. Theorem 3 shows that Π implies $\neg_p\langle author\rangle$ and $\neg_c\langle title\rangle$, but neither $\neg_p\langle journal\rangle$ nor $\neg_c\langle author, journal\rangle$.*

Corollary 1 *Implication of possible and certain keys, as well as implication of strong and weak anti-keys, can be decided in linear time.*

3.4 Axiomatization

Let $\Sigma \cup \{\varphi\}$ denote a set of possible and certain keys over (T, T_S). Let $\Sigma^* = \{\varphi \mid \Sigma \models \varphi\}$ denote the *semantic closure* of Σ. In order to determine the semantic closure, one can utilize a syntactic approach by applying *inference rules* of the form

$$\frac{\text{premise}}{\text{conclusion}}\ \text{condition},$$

where rules without a premise are called *axioms*. For a set $\mathfrak{R}$ of inference rules let $\Sigma \vdash_{\mathfrak{R}} \varphi$ denote the *inference* of φ from Σ by $\mathfrak{R}$. That is, there is some sequence $\sigma_1, \ldots, \sigma_n$ such that $\sigma_n = \varphi$ and every σ_i is an element of Σ or is the conclusion that results from an application of an inference rule in $\mathfrak{R}$ to some premises in $\{\sigma_1, \ldots, \sigma_{i-1}\}$. Let $\Sigma^+_{\mathfrak{R}} = \{\varphi \mid \Sigma \vdash_{\mathfrak{R}} \varphi\}$ denote the *syntactic closure* of Σ under inferences by $\mathfrak{R}$. $\mathfrak{R}$ is *sound* (*complete*) if for every table schema (T, T_S) and for every set Σ we have $\Sigma^+_{\mathfrak{R}} \subseteq \Sigma^*$ ($\Sigma^* \subseteq \Sigma^+_{\mathfrak{R}}$). The (finite) set $\mathfrak{R}$ is said to be a (finite) *axiomatization* if $\mathfrak{R}$ is both sound and complete.

Theorem 4 *The following axioms are sound and complete for implication of possible and certain keys.*

$$\textit{p-Extension:}\quad \frac{p\langle X\rangle}{p\langle XY\rangle} \qquad \textit{c-Extension:}\quad \frac{c\langle X\rangle}{c\langle XY\rangle}$$

$$\textit{Weakening:}\quad \frac{c\langle X\rangle}{p\langle X\rangle} \qquad \textit{Strengthening:}\quad \frac{p\langle X\rangle}{c\langle X\rangle}\, X \subseteq T_S$$

Example 6 *Let $T = \{title, author, journal\}$ be our table schema, $T_S = \{title, author\}$ our NFS, and let Σ consist of $p\langle journal\rangle$, $c\langle title, journal\rangle$ and $c\langle author, journal\rangle$. Then $p\langle author, journal\rangle$ can be inferred from Σ by a single application of p-Extension to $p\langle journal\rangle$, or by a single application of Weakening to $c\langle author, journal\rangle$.*

3.5 Minimal Covers

A *cover* of Σ is a set Σ' where every element is implied by Σ and which implies every element of Σ. Hence, a cover is just a representation. Minimal representations of constraint sets are of particular interest in database practice. Firstly, they justify the use of valuable resources, for example, by limiting the validation of constraints to those necessary. Secondly, people find minimal representations easier to work with than non-minimal ones. For the class of possible and certain keys, a unique minimal representation exists.

Definition 4 (Minimal/Redundant Keys)
We say that

i) $p\langle X\rangle \in \Sigma$ *is* non-minimal *if* $\Sigma \models p\langle Y\rangle$ *for some* $Y \subset X$ *or* $\Sigma \models c\langle X\rangle$*, and*

ii) $c\langle X\rangle \in \Sigma$ *is* non-minimal *if* $\Sigma \models c\langle Y\rangle$ *for some* $Y \subset X$*.*

We call a key σ with $\Sigma \models \sigma$ minimal *iff σ is not non-minimal. We call $\sigma \in \Sigma$* redundant *iff $\Sigma \setminus \{\sigma\} \models \sigma$, and* non-redundant *otherwise. We call Σ* minimal *(*non-redundant*) iff all keys in Σ are minimal (non-redundant), and* non-minimal *(*redundant*) otherwise.*

Due to the logical equivalence of $p\langle X\rangle$ and $c\langle X\rangle$ for $X \subseteq T_S$, certain keys can be both minimal and redundant while possible keys can be both non-minimal and non-redundant.

Lemma 1 *The set Σ_{min} of all minimal possible and certain keys w.r.t. Σ is a non-redundant cover of Σ.*

Corollary 2 *The minimal cover from Lemma 1 is the only minimal cover of* Σ.

We may hence talk about *the* minimal cover of Σ.

Example 7 *Let* $T = \{title, author, journal\}$ *be our table schema,* $T_S = \{title, author\}$ *our NFS, let* Σ' *consist of* $p\langle journal\rangle$, $c\langle title, journal\rangle$, $c\langle author, journal\rangle$, $p\langle author, journal\rangle$ *and* $c\langle title, author, journal\rangle$. *The minimal cover* Σ *of* Σ' *consists of* $p\langle journal\rangle$, $c\langle title, journal\rangle$ *and* $c\langle author, journal\rangle$.

Definition 5 (Minimal Anti-Key)
We say that

i) a strong anti-key $\neg_p\langle X\rangle$ *is* non-maximal *if* $\neg\langle Y\rangle$ *with* $X \subset Y$ *is an anti-key implying* $\neg_p\langle X\rangle$ *on* (T, T_S), *and*

ii) a weak anti-key $\neg_c\langle X\rangle$ *is* non-maximal *if* $\neg\langle Y\rangle$ *with* $X \subset Y$ *is an anti-key or* $\neg_p\langle X\rangle$ *is a strong anti-key.*

Again we call anti-keys maximal *unless they are non-maximal. We shall denote the set of all maximal strong anti-keys by* $\mathcal{A}^s_{max}$, *the set of all maximal weak anti-keys by* $\mathcal{A}^w_{max}$ *and their disjoint union by* $\mathcal{A}_{max}$.

3.6 Key Discovery

Our next goal is to find all certain and/or possible keys that hold for a given table I over (T, T_S). If T_S is not given it is trivial to find a maximal T_S. In general, the discovery of constraints from data is an important task: the constraints represent semantic information about the data that can help with database design and administration, in data exchange and integration; some meaningful constraints may have not been specified, or some meaningless constraints may hold accidentally but could be exploited in query optimization.

Keys can be discovered from total tables by computing the agree sets of all pairs of distinct tuples, and then computing the transversals for their complements [14]. On general tables we distinguish between strong and weak agree sets, motivated by our notions of strong and weak similarity.

Definition 6 (Agree Set)
Given two tuples t, t' *over* T, *the* weak (strong) agree set *of* t, t' *is the (unique) maximal subset* $X \subseteq T$ *such that* t, t' *are weakly (strongly) similar on* X. *Given a table* I *over* T, *we denote by* $\mathcal{AG}^w(I), \mathcal{AG}^s(I)$ *the set of all maximal agree sets of distinct tuples in* I:

$$\mathcal{AG}^w(I) := \max\{X \mid \exists t \neq t' \in I.t[X] \sim_w t'[X]\}$$
$$\mathcal{AG}^s(I) := \max\{X \mid \exists t \neq t' \in I.t[X] \sim_s t'[X]\}$$

We shall simply write $\mathcal{AG}^w, \mathcal{AG}^s$ *when* I *is clear from the context.*

Complements and transversals are standard notions for which we now introduce notation.

Definition 7 (complement and transversal)
Let X be a subset of T and $\mathcal{S}$ a set of subset of T. We use the following notation for complements and transversals:

$$\begin{aligned} \overline{X} &:= T \setminus X \\ \overline{\mathcal{S}} &:= \{\overline{X} \mid X \in \mathcal{S}\} \\ Tr(\mathcal{S}) &:= \min\{Y \subseteq T \mid \forall X \in \mathcal{S}.\, Y \cap X \neq \emptyset\} \end{aligned}$$

Certain (possible) keys that hold in I are the transversals of the complements for all weak (strong) agree sets in I.

Theorem 5 *Let I be a table over (T, T_S), and Σ_I the set of all certain and/or possible keys that hold on I. Then*

$$\Sigma := \{c\,\langle X\rangle \mid X \in Tr(\overline{\mathcal{AG}^w})\} \cup \{p\,\langle X\rangle \mid X \in Tr(\overline{\mathcal{AG}^s})\}$$

is a cover of Σ_I.

Example 8 *Let I denote Table 1 over table schema $T = \{title, author, journal\}$. Then $T_S = \{title, author\}$ can be easily computed. Then $\mathcal{AG}^w(I)$ consists of $\{title,author\}$ and $\{journal\}$, with complements $\{journal\}$ and $\{title,author\}$ in $\overline{\mathcal{AG}^w}$, and transversals $\{title,journal\}$ and $\{author,journal\}$. $\mathcal{AG}^s(I)$ consists of $\{title,author\}$, with complement $\{journal\}$ in $\overline{\mathcal{AG}^s}$, and transversal $\{journal\}$. Therefore, a cover of the set of possible and certain keys that holds on I consists of $c\,\langle title, journal\rangle$, $c\,\langle author, journal\rangle$, and $p\,\langle journal\rangle$.*

4 Armstrong Tables

Armstrong tables are widely regarded as a user-friendly, exact representation of abstract sets of constraints [5, 17, 22, 36]. For a class $\mathcal{C}$ of constraints and a set Σ of constraints in $\mathcal{C}$, a $\mathcal{C}$-Armstrong table I for Σ satisfies Σ and violates all the constraints in $\mathcal{C}$ not implied by Σ. Therefore, given an Armstrong table I for Σ the problem of deciding for an arbitrary constraint in $\mathcal{C}$ whether Σ implies φ reduces to the problem of verifying whether φ holds on I. The ability to compute an Armstrong table for Σ provides us with a data sample that is a perfect summary of the semantics embodied in Σ. Unfortunately, classes $\mathcal{C}$ cannot be expected at all to enjoy Armstrong tables. That is, there are sets Σ for which no $\mathcal{C}$-Armstrong table exists [17]. Classically, keys and even functional dependencies enjoy Armstrong relations [5, 36]. However, for possible and certain keys under `NOT NULL` constraints the situation is more involved. Nevertheless, we will characterize when Armstrong tables exist, and establish structural and computational properties for them.

4.1 Definition and Some Examples

Definition 8 (Armstrong table)
An instance I over (T, T_S) is a pre-Armstrong table *for (T, T_S, Σ) if for every key σ*

over T, σ holds on I iff $\Sigma \models \sigma$. Furthermore we call I an Armstrong table *if it is a pre-Armstrong table and for every* `NULL` *attribute $A \in T \setminus T_S$ there exists a tuple $t \in I$ with $t[A] = \bot$.*

As the following examples show, there are cases where no pre-Armstrong tables exist, as well as cases where a pre-Armstrong table but not Armstrong tables exist.

Example 9 (pre-Armstrong but no Armstrong)
Let $(T, T_S, \Sigma) = (AB, A, \{c\langle B\rangle\})$. One may verify that the following is indeed a pre-Armstrong table:

A	B
0	0
0	1

Now let I be an instance over (T, T_S, Σ) with $t \in I$ such that $t[B] = \bot$. Then the existence of any other tuple $t' \in I$ with $t' \neq t$ would violate $c\langle B\rangle$, so $I = \{t\}$. But that means $c\langle A\rangle$ holds on I even though $\Sigma \not\models c\langle A\rangle$, so I is not pre-Armstrong.

Example 10 (no pre-Armstrong table)
Let $(T, T_S) = (ABCD, \emptyset)$ and

$$\Sigma = \{c\langle AB\rangle, c\langle CD\rangle, p\langle AC\rangle, p\langle AD\rangle, p\langle BC\rangle, p\langle BD\rangle\}$$

Then a pre-Armstrong table I must disprove the certain keys $c\langle AC\rangle$, $c\langle AD\rangle$, $c\langle BC\rangle$, $c\langle BD\rangle$ while respecting the possible keys $p\langle AC\rangle$, $p\langle AD\rangle$, $p\langle BC\rangle$, $p\langle BD\rangle$. In each of these four cases, we require two tuples t, t' which are weakly but not strongly similar on the corresponding key sets (e.g. $t[AC] \sim_w t'[AC]$ but $t[AC] \not\sim_s t'[AC]$). This is only possible when t or t' are $\bot$ on one of the key attributes. This ensures the existence of tuples $t_{AC}, t_{AD}, t_{BC}, t_{BD}$ with

$$t_{AC}[A] = \bot \ \vee \ t_{AC}[C] = \bot, \quad t_{AD}[A] = \bot \ \vee \ t_{AD}[D] = \bot,$$
$$t_{BC}[B] = \bot \ \vee \ t_{BC}[C] = \bot, \quad t_{BD}[B] = \bot \ \vee \ t_{BD}[D] = \bot$$

If $t_{AC}[A] \neq \bot$ and $t_{AD}[A] \neq \bot$ it follows that $t_{AC}[C] = \bot$ and $t_{AD}[D] = \bot$. But this means $t_{AC}[CD] \sim_w t_{AD}[CD]$, contradicting $c\langle CD\rangle$[1]. Hence there exists $t_A \in \{t_{AC}, t_{AD}\}$ with $t_A[A] = \bot$. Similarly we get $t_B \in \{t_{BC}, t_{BD}\}$ with $t_B[B] = \bot$. Then $t_A[AB] \sim_w t_B[AB]$ contradicting $c\langle AB\rangle$.

The examples in this section provide strong motivation to investigate when exactly Armstrong tables do exist.

4.2 Structural Characterization

Definition 9 ($\bot$-base)
Let $t_1, \ldots, t_n$ be tuples over T. The $\bot$-base of $t_1, \ldots, t_n$ *consists of all attributes where any tuple is $\bot$:*

$$\bot\text{-}base(t_1, \ldots, t_n) := \{A \in T \mid t_1[A] = \bot \vee \ldots \vee t_n[A] = \bot\}$$

[1] Even if $t_{AC} = t_{AD}$, see Lemma 2.

Lemma 2 *Let I be an instance over (T, T_S) with $|I| \geq 2$. Then I violates $c\langle \bot\text{-base}(t,t')\rangle$ for every $t, t' \in I$.*

If $\Sigma \vDash c\langle\emptyset\rangle$ there is an Armstrong table containing a single tuple only. Otherwise a pre-Armstrong table must contain at least two tuples, so Lemma 2 applies.

Theorem 6 *Let I be an instance over (T, T_S) such that Σ holds on I. Then I is a pre-Armstrong table of Σ iff*

i) for every strong anti-key $\neg_p\langle X\rangle \in \mathcal{A}^s_{max}$ there exist distinct tuples $t, t' \in I$ with $t[X] \sim_s t'[X]$, and

ii) for every weak anti-key $\neg_c\langle X\rangle \in \mathcal{A}^w_{max}$ there exist distinct tuples $t, t' \in I$ with $t[X] \sim_w t'[X]$.

Definition 10 (cross-union)
Let $\mathcal{V}, \mathcal{W} \subseteq \mathcal{P}(T)$ be two sets of sets. The cross-union *of $\mathcal{V}$ and $\mathcal{W}$ is defined as*

$$\mathcal{V} \bowtie \mathcal{W} := \{V \cup W \mid V \in \mathcal{V}, W \in \mathcal{W}\}$$

We abbreviate the cross-union of a set $\mathcal{W}$ with itself by $\mathcal{W}^{\times 2} := \mathcal{W} \bowtie \mathcal{W}$.

Theorem 7 *Let $\Sigma \nvDash c\langle\emptyset\rangle$. There exists a pre-Armstrong table for (T, T_S, Σ) iff there exists a set $\mathcal{W} \subseteq \mathcal{P}(T \setminus T_S)$ with the following properties:*

i) Every element of $\mathcal{W}^{\times 2}$ forms a weak anti-key.

ii) For every maximal weak anti-key $\neg_c\langle X\rangle \in \mathcal{A}^w_{max}$ there exists $Y \in \mathcal{W}^{\times 2}$ with $Y \cap X' \neq \emptyset$ for every possible key $p\langle X'\rangle \in \Sigma$ with $X' \subseteq X$.

There exists an Armstrong table for (T, T_S, Σ) iff i) and ii) hold as well as

iii) $\bigcup \mathcal{W} = T \setminus T_S$.

In the following we shall assume w.l.o.g. that all attributes have integer domains.

Construction 1 (Armstrong Table)
Let $\mathcal{W} \subseteq \mathcal{P}(T \setminus T_S)$ satisfy conditions i) and ii) of Theorem 7. We construct an instance I over (T, T_S) as follows.

I) For every strong anti-key $\neg_p\langle X\rangle \in \mathcal{A}^s_{max}$ add tuples $t^s_X, t^{s\prime}_X$ to I with

$$\begin{array}{ll} t^s_X[X] = (i, \ldots, i) & t^{s\prime}_X[X] = (i, \ldots, i) \\ t^s_X[T \setminus X] = (j, \ldots, j) & t^{s\prime}_X[T \setminus X] = (k, \ldots, k) \end{array}$$

where i, j, k are distinct integers not used previously.

II) For every weak anti-key $\neg_c\langle X\rangle \in \mathcal{A}^w_{max}$ we add tuples $t^w_X, t^{w\prime}_X$ to I with

$$t^w_X[X \setminus Y_1] = (i, \ldots, i) \qquad t^{w\prime}_X[X \setminus Y_2] = (i, \ldots, i)$$
$$t^w_X[X \cap Y_1] = (\bot, \ldots, \bot) \quad t^{w\prime}_X[X \cap Y_2] = (\bot, \ldots, \bot)$$
$$t^w_X[T \setminus X] = (j, \ldots, j) \qquad t^{w\prime}_X[T \setminus X] = (k, \ldots, k)$$

where $Y_1, Y_2 \in \mathcal{W}$ meet the condition for $Y = Y_1 \cup Y_2$ in ii) of Theorem 7, and i, j, k are distinct integers not used previously.

III) If condition iii) of Theorem 7 also holds for $\mathcal{W}$, then for every $A \in T \setminus T_S$ with $A \notin \bigcup\{\bot\text{-base}(t) \mid t \in I\}$ we add a tuple t_A to I with

$$t_A[T \setminus A] = (i, \ldots, i) \quad t_A[A] = \bot$$

where i is an integer not used previously.

Theorem 8 *Let $\Sigma \not\models c\langle\emptyset\rangle$ and I constructed via Construction 1. Then I is a pre-Armstrong table over (T, T_S, Σ). If condition iii) of Theorem 7 holds for $\mathcal{W}$, then I is an Armstrong table.*

The characterization of Theorem 7 is difficult to test in general, due to the large number of candidate sets $\mathcal{W}$. However, there are some cases where this becomes easy.

Corollary 3 *Let $\Sigma \not\models c\langle\emptyset\rangle$.*

i) If $\Sigma \not\models c\langle X\rangle$ for every $X \subseteq T \setminus T_S$ then there exists an Armstrong table for (T, T_S, Σ).

ii) If $\Sigma \models c\langle X\rangle$ for any $X \subseteq T \setminus T_S$ with $|X| \leq 2$ then there does not exist an Armstrong table for (T, T_S, Σ).

Example 11 *Consider our running example where $T = \{article, author, journal\}$, $T_S = \{article, author\}$ and Σ consists of the certain key $c\langle article, journal\rangle$, the certain key $c\langle author, journal\rangle$ and the possible key $p\langle journal\rangle$. This gives us the maximal strong and weak anti-keys*

$$\mathcal{A}_{max} = \{\neg_p\langle author, article\rangle, \neg_c\langle journal\rangle\}$$

with the set $\mathcal{W} = \{journal\}$ meeting the conditions of Theorem 7. Now Construction 1 produces the Armstrong table

article	*author*	*journal*
0	0	0
0	0	1
1	1	$\bot$
2	2	$\bot$

.

In this specific case, we can remove either the third or the fourth tuple. While those two tuples have weak agree set $\{journal\}$, both of them have the same weak agree set with the first and the second tuple, too. After removal of the third or fourth tuple and suitable substitution we obtain Table 1.

4.3 Computation

Deciding whether an Armstrong table exists appears to be computationally demanding. Proving that it is actually NP-hard appears to be difficult as well. A key problem here is that the characterization of Theorem 7 requires the set of maximal weak anti-keys, which can be exponential in the size of Σ, and vice versa. It is thus not even clear if the problem lies in NP. Hence we will only show NP-hardness for the *key/anti-key satisfiability problem*, which is closely related, and therefore treat it as circumstantial evidence for the difficulty of the Armstrong existence problem. We then develop a worst-case exponential algorithm to decide Armstrong existence (or key/anti-key satisfiability), which nevertheless seems to be efficient in practice.

4.3.1 NP-completeness

Problem 1 (key/anti-key satisfiability) *Given a schema (T, T_S, Σ) and a set $\mathcal{A}^w \subseteq \mathcal{P}(T)$ of weak anti-keys, does there exist a table I over (T, T_S, Σ) such that every element of $\mathcal{A}^w$ is a weak anti-key for I?*

Adding strong anti-keys to the key/anti-key satisfiability problem does not make it harder: If a strong anti-key directly contradicts a certain or possible key this is easily detected. Otherwise any table I for which Σ and $\mathcal{A}^w$ hold can easily be extended by Construction 1 to satisfy strong anti-keys as well. The pre-Armstrong existence problem can be reduced (though not in polynomial time) to the key/anti-key satisfiability problem by computing the set of maximal weak anti-keys w.r.t. Σ. We will follow this approach in Algorithm 1, so by showing key/anti-key satisfiability to be NP-complete, we show at least that any attempt to decide Armstrong existence in polynomial time will likely need to take a very different route. As $\mathcal{W}$ can be exponential in the size of Σ, any such approach must not compute $\mathcal{W}$ at all.

The NP-hard problem [40] we reduce to the key/anti-key satisfiability problem is *monotone 1-in-3 SAT*.

Problem 2 (monotone 1-in-3 SAT) *Given a set of 3-clauses without negations, does there exist a truth assignment such that every clause contains exactly one true literal?*

Theorem 9 *The key/anti-key satisfiability problem is NP-complete.*

4.3.2 Algorithms

Maximal anti-keys can be constructed using transversals.

Lemma 3 *Let Σ^P, Σ^C be the sets of possible and certain keys in Σ. For readability we will identify attribute sets with the keys or anti-keys induced by them. Then*

$$\begin{aligned}\mathcal{A}^s_{max} = \{\ X \in \overline{Tr(\Sigma^C \cup \Sigma^P)} \mid \\ \neg(X \subseteq T_S \wedge \exists Y \in \mathcal{A}^w_{max}.\ X \subset Y)\ \}\end{aligned}$$

$$\mathcal{A}^w_{max} = \overline{Tr(\Sigma^C \cup \{X \in \Sigma^P \mid X \subseteq T_S\})} \setminus \mathcal{A}^s_{max}$$

In order to use Lemma 3 to compute $\mathcal{A}^s_{max}$ and $\mathcal{A}^w_{max}$, we observe that anti-keys for which strictly larger weak anti-keys exist can never be setwise maximal weak anti-keys. Hence the set of all maximal weak anti-keys can be computed as (again identifying sets with their induced weak anti-keys)

$$\mathcal{A}^w_{max} = \overline{Tr(\Sigma^C \cup \{X \in \Sigma^P \mid X \subseteq T_S\})} \setminus \overline{Tr(\Sigma^C \cup \Sigma^P)}$$

The difficult part in deciding existence of (and then computing) an Armstrong table given (T, T_S, Σ) is to check existence of (and construct) a set $\mathcal{W}$ meeting the conditions of Theorem 7, in cases where Corollary 3 does not apply. Blindly testing all subsets of $\mathcal{P}(T \setminus T_S)$ becomes infeasible as soon as $T \setminus T_S$ contains more than 4 elements (for 5 elements, up to $2^{32} \approx 4,000,000,000$ sets would need to be checked). To ease discussion we will rephrase condition ii).

Definition 11 (support)
Let $\mathcal{W} \subseteq \mathcal{P}(T)$. We say that

- $\mathcal{W}$ supports $Y \subseteq T$ *if* $Y \subseteq Z$ *for some* $Z \in \mathcal{W}$.
- $\mathcal{W}$ $\vee$-supports $\mathcal{V} \subseteq \mathcal{P}(T)$ *if* $\mathcal{W}$ *supports* some $Y \in \mathcal{V}$.
- $\mathcal{W}$ $\wedge\vee$-supports $\mathcal{T} \subseteq \mathcal{P}(\mathcal{P}(T))$ *if* $\mathcal{W}$ $\vee$*-supports* every $\mathcal{V} \in \mathcal{T}$.

We write $Y \Subset \mathcal{W}$ to indicate that $\mathcal{W}$ supports Y.

Lemma 4 *Let $\mathcal{T}_X$ be the transversal set in $T \setminus T_S$ of all possible keys that are subsets of X, and $\mathcal{T}$ the set of all such transversals for all maximal weak antikeys:*

$$\mathcal{T}_X := Tr(\{X' \cap (T \setminus T_S) \mid p\,\langle X' \rangle \in \Sigma_{min} \wedge X' \subseteq X\})$$
$$\mathcal{T} := \{\mathcal{T}_X \mid X \in \mathcal{A}^w_{max}\}$$

Then condition ii) of Theorem 7 can be rephrased as follows:

ii') $\mathcal{W}^{\times 2}$ $\wedge\vee$-supports $\mathcal{T}$.

We propose the following: For each $\mathcal{T}_X \in \mathcal{T}$ and each minimal transversal $t \in \mathcal{T}_X$ we generate all non-trivial bipartitions $\mathcal{B}_\mathcal{X}$ (or just a trivial partition for transversals of cardinality < 2). We then add to $\mathcal{W}$ one such bi-partition for every $\mathcal{T}_X$ to ensure condition ii), and combine them with all single-attribute sets $\{A\} \subseteq T \setminus T_S$ to ensure condition iii). This is done for every possible combination of bipartitions until we find a set $\mathcal{W}$ that meets condition i), or until we have tested them all. We then optimize this strategy as follows: If a set $\mathcal{P}_X$ is already $\vee$-supported by $\mathcal{W}^{\times 2}$ (which at the time of checking will contain only picks for some sets $\mathcal{T}_X$), we may remove $\mathcal{P}_X$ from further consideration, as long as we keep all current picks in $\mathcal{W}$. In particular, since all single-attribute subsets of $T \setminus T_S$ are added to $\mathcal{W}$, we may ignore all $\mathcal{P}_X$ containing a set of size 2 or less. We give the algorithm thus developed in pseudo-code as Algorithm 1.

Algorithm 1 Armstrong-Set

Input: T, T_S, Σ

Output: $\mathcal{W} \subseteq \mathcal{P}(T \setminus T_S)$ meeting conditions i) to iii) of Theorem 7 if such $\mathcal{W}$ exists, $\perp$ otherwise

1: **if** $\nexists c\langle X\rangle \in \Sigma$ with $X \subseteq T \setminus T_S$ **then**
2: **return** $\{T \setminus T_S\}$
3: **if** $\exists c\langle X\rangle \in \Sigma$ with $X \subseteq T \setminus T_S$ and $|X| \leq 2$ **then**
4: **return** $\perp$
5: $\mathcal{W} := \{\{A\} \mid A \in T \setminus T_S\}$
6: $\mathcal{A}^w_{max} := \overline{Tr(\Sigma^C \cup \{X \in \Sigma^P \mid X \subseteq T_S\})} \setminus \overline{Tr(\Sigma^C \cup \Sigma^P)}$
7: $\mathcal{T} := \{\ Tr(\{X' \cap (T \setminus T_S) \mid p\langle X'\rangle \in \Sigma_{min} \wedge X' \subseteq X\}) \mid X \in \mathcal{A}^w_{max}\ \}$
8: $\mathcal{T} := \mathcal{T} \setminus \{\mathcal{T}_X \in \mathcal{T} \mid \exists Y \in \mathcal{T}_X.\ |Y| \leq 2\}$
9: **return** Extend-Support$(\mathcal{W}, \mathcal{T})$

Subroutine Extend-Support$(\mathcal{W}, \mathcal{T})$

Input: $\mathcal{W} \subseteq \mathcal{P}(T \setminus T_S)$ meeting conditions i) and iii) of Theorem 7, $\mathcal{T} \subseteq \mathcal{P}(\mathcal{P}(T \setminus T_S))$

Output: $\mathcal{W}' \supseteq \mathcal{W}$ meeting conditions i) and iii) of Theorem 7 such that $\mathcal{W}'^{\times 2}$ $\wedge\vee$-supports $\mathcal{T}$ if such $\mathcal{W}'$ exists, $\perp$ otherwise

10: **if** $\mathcal{T} = \emptyset$ **then**
11: **return** $\mathcal{W}$
12: $\mathcal{T} := \mathcal{T} \setminus \{\mathcal{T}_X\}$ for some $\mathcal{T}_X \in \mathcal{T}$
13: **if** $\mathcal{W}^{\times 2}$ $\vee$-supports $\mathcal{T}_X$ **then**
14: **return** Extend-Support$(\mathcal{W}, \mathcal{T})$
15: **for all** $Y \in \mathcal{T}_X$ **do**
16: **for all** non-trivial bipartitions $Y = Y_1 \cup Y_2$ **do**
17: **if** $(\mathcal{W} \cup \{Y_1, Y_2\})^{\times 2}$ contains no certain key **then**
18: $\mathcal{W}' :=$ Extend-Support$(\mathcal{W} \cup \{Y_1, Y_2\}, \mathcal{T})$
19: **if** $\mathcal{W}' \neq \perp$ **then**
20: **return** $\mathcal{W}'$
21: **return** $\perp$

Lemma 5 *Algorithm Armstrong-Set is correct.*

Example 12 *Let* $(T, T_S) = (ABCDE, \emptyset)$ *and*

$$\Sigma = \left\{ \begin{array}{l} p\langle A\rangle, p\langle B\rangle, p\langle CD\rangle, \\ c\langle ABE\rangle, c\langle ACE\rangle, c\langle ADE\rangle, c\langle BCE\rangle \end{array} \right\}$$

Neither condition of Corollary 3 is met, so Algorithm Armstrong-Set initializes/computes $\mathcal{W}$, $\mathcal{A}^w_{max}$ *and* $\mathcal{T}$ *as*

$$\begin{aligned} \mathcal{W} &= \{A, B, C, D, E\} \\ \mathcal{A}^w_{max} &= \{ABCD, AE, BDE, CDE\} \\ \mathcal{T} &= \{\{ABC, ABD\}\} \end{aligned}$$

Then, in method Extend-Support, $\mathcal{T}_x = \{ABC, ABD\}$ which is not $\vee$-supported by $\mathcal{W}^{\times 2}$. The non-trivial bi-partitions (Y_1, Y_2) of ABC are

$$\{(A, BC), (AC, B), (AB, C)\}.$$

None of these are suitable for extending $\mathcal{W}$, as the extension $(\mathcal{W} \cup \{Y_1, Y_2\})^{\times 2}$ contains the certain keys BCE, ACE and ABE, respectively. The non-trivial bi-partitions of the second set ABD are $\{(A, BD), (AD, B), (AB, D)\}$. While (AD, B) and (AB, D) are again unsuitable, (A, BD) can be used to extend $\mathcal{W}$ to the Armstrong set

$$\begin{aligned}\mathcal{W}' &= \textit{Extend-Support}(\{A, BD, C, E\}, \emptyset)\\ &= \{A, BD, C, E\}\end{aligned}$$

If we add $c\,\langle BDE\rangle$ to the sample schema, then (A, BD) becomes unsuitable as well, so that no Armstrong set exists.

5 Some Extremal Combinatorics

Database administrators find it generally useful to know how complex the maintenance of their database can grow. Here, we establish such knowledge for the maintenance of possible and certain keys. We provide answers to basic questions concerning the maximum cardinality that non-redundant families of possible and certain keys can have, and which families attain this cardinality. A characterization of non-redundant families enables us to apply techniques from extremal set theory to answer our questions. The main result is interesting from a combinatorial perspective itself, as it generalizes the famous theorem by Sperner [42].

In this section we write $[n] := \{1, \ldots, n\}$ instead of $T = \{A_1, \ldots, A_n\}$, and A instead of T_S. For $X \subseteq [n]$, we use the notations $X^{(i)} := \{Y \subseteq X : |Y| = i\}$ and $X^{(\leq i)} := \{Y \subseteq X : |Y| \leq i\}$. A family $\mathcal{A} \subseteq [n]^{(\leq n)}$ is an *antichain* (or *Sperner family*) if $X \not\subseteq Y$ for all distinct $X, Y \in \mathcal{A}$. For a set Σ of possible and certain keys over $[n]$, define $\mathcal{F} := \{X \mid c\,\langle X\rangle \in \Sigma\}$ and $\mathcal{G} := \{X \mid p\,\langle X\rangle \in \Sigma\}$.

Theorem 10 *Σ is non-redundant if and only if the following conditions are satisfied (1) $\mathcal{F}$ is an antichain, (2) $\mathcal{G}$ is an antichain, (3) $\forall\, F \in \mathcal{F}, G \in \mathcal{G} :\ F \not\subseteq G$, and (4) $\forall\, F \in \mathcal{F}, G \in \mathcal{G} :\ G \not\subseteq F \cap A$.*

The problem we study reads as follows. Given non-negative integers n, k with $k \leq n$ and a set $A \subseteq [n]$, find all pairs $(\mathcal{F}, \mathcal{G}) \in [n]^{(\leq k)} \times [n]^{(\leq k)}$ that satisfy conditions (1)–(4) and maximize $|\mathcal{F} \cup \mathcal{G}|$. Note that $\mathcal{F}$ and $\mathcal{G}$ must be disjoint because of (3). In what follows, a complete solution is given.

First, we briefly discuss the case when $A = [n]$. In this case, (1)–(4) are satisfied if and only if $\mathcal{F} \dot\cup \mathcal{G} \subseteq [n]^{(\leq k)}$ is an antichain. If $k > n/2$, then by Sperner's Theorem [42] $|\mathcal{F} \cup \mathcal{G}|$ attains its maximum if and only if $\mathcal{F}$ and $\mathcal{G}$ form a partition of $[n]^{(\lfloor n/2 \rfloor)}$ or of $[n]^{(\lceil n/2 \rceil)}$, respectively. If $k \leq n/2$, then $|\mathcal{F} \cup \mathcal{G}|$ is maximized if and only if $\mathcal{F}$ and $\mathcal{G}$ form a partition of $[n]^{(k)}$. This is well-known and follows from the original proof of Sperner's Theorem [42]. In the sequel, we will assume that $0 \leq |A| < n$. Note that the bound (1) below does not hold when $A = [n]$ for even n and $k > n/2$.

Theorem 11 *Let n, a, k be non-negative integers with $n \geq 2$, $a < n$ and $k \leq n$, and let $A \subseteq [n]$ with $|A| = a$. Furthermore, let $\mathcal{F}, \mathcal{G} \subseteq [n]^{(\leq k)}$ be antichains such that $F \not\subseteq G$ and $G \not\subseteq F \cap A$ for all $F \in \mathcal{F}$ and $G \in \mathcal{G}$. Then*

$$|\mathcal{F} \cup \mathcal{G}| \leq \binom{n+1}{m} - \binom{a}{m-1}, \tag{1}$$

where $m := \min\{k, \lfloor n/2 \rfloor + 1\}$. This bound is the best possible, and equality is attained if and only if

(i) $\mathcal{F} \dot{\cup} \mathcal{G} = [n]^{(m)} \cup ([n]^{(m-1)} \setminus A^{(m-1)})$, *or*

(ii) $\mathcal{F} \dot{\cup} \mathcal{G} = [n]^{(m-1)} \cup ([n]^{(m-2)} \setminus A^{(m-2)})$, *where n is even, $k > n/2$, and $a \leq n/2 - 2$ or $a = n - 1$*

Furthermore, for $k > 1$, (i) implies

$$[n]^{(m)} \setminus A^{(m)} \subseteq \mathcal{F} \quad \textit{and} \quad [n]^{(m-1)} \setminus A^{(m-1)} \subseteq \mathcal{G}$$

while (ii) implies

$$[n]^{(m-1)} \setminus A^{(m-1)} \subseteq \mathcal{F} \quad \textit{and} \quad [n]^{(m-2)} \setminus A^{(m-2)} \subseteq \mathcal{G}\ .$$

Note that $[n]^{(m)}$ and $[n]^{(m-1)}$ are the two largest disjoint anti-chains over $[n]^{(\leq k)}$. To prevent $F \subseteq G$, F must contain the larger sets. To prevent $G \subseteq F \cap A$ we drop small subsets of A. The remaining subsets of A may be attributed to either F or G as they do not conflict with others. This reflects the equivalence of certain/possible keys over `NOT NULL` attributes. While case (i) always presents a non-redundant family of maximum cardinality, case (ii) occurs because for even n and large k, an additional pair of largest disjoint anti-chains exists: $[n]^{(m-1)}$ and $[n]^{(m-2)}$. The difference between the resulting sets lies in the size of $A^{(m-1)}$ and $A^{(m-2)}$, leading to the conditions for a. The trivial case $n = 1$ is excluded above to avoid certain technicalities in its proof.

Example 13 *For table schema $(T = \{A, B, C, D\}, T_S = \{A, B\})$, or $n = 4$ and $a = 2$, the maximum cardinality of a non-redundant family of possible and certain keys is nine. The bound is attained by the set Σ where*

$$\Sigma = \left\{ \begin{array}{l} c\langle A, B, C\rangle, c\langle A, B, D\rangle, c\langle A, C, D\rangle, c\langle B, C, D\rangle, \\ p\langle A, C\rangle, p\langle B, C\rangle, p\langle A, D\rangle, p\langle B, D\rangle, c\langle B, D\rangle \end{array} \right\}.$$

The associated powerset lattice is shown in Figure 1, where the top marked circles represent the four certain keys and the bottom marked circles represent the five possible keys above.

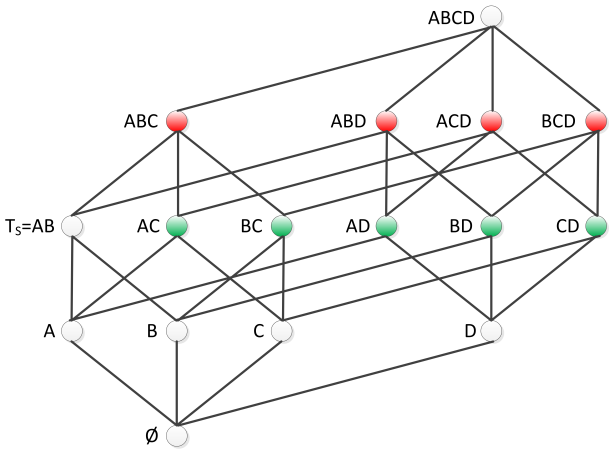

Figure 1: Maximum Non-redundant Family of Keys

6 Enforcing Certain Keys

Index structures are required to enforce certain keys efficiently in practice. This problem is non-trivial as weak similarity is not transitive. Hence, classic indices will not work directly. Nevertheless we will present an index scheme, based on multiple classical indices, which allows us to check certain keys efficiently, provided there are few nullable key attributes. While an efficient index scheme seems elusive for larger sets of nullable attributes, we expect that most certain keys in practice have only few nullable attributes.

Definition 12 (certain-key-index)
Let (T, T_S) be a table schema and $X \subseteq T$. A certain-key-index *for $c\langle X\rangle$ consists of a collection of indices $\mathfrak{I}_Y$ on subsets Y of X which include all* `NOT NULL` *attributes:*

$$\mathfrak{I}_{c\langle X\rangle} := \{\mathfrak{I}_Y \mid X \cap T_S \subseteq Y \subseteq X\}$$

Here we treat $\perp$ as regular value for the purpose of indexing, i.e., we do index tuples with $\perp$ "values". When indexing a table I, each tuple in I is indexed in each $\mathfrak{I}$.

Obviously, $|\mathfrak{I}_{c\langle X\rangle}| = 2^n$, where $n := |X \setminus T_S|$, which makes maintenance efficient only for small n. When checking if a tuple exists that is weakly similar to some given tuple, we only need to consult a single index, but within that index we must perform up to 2^n lookups.

Theorem 12 *Let t be a tuple on (T, T_S) and $\mathfrak{I}_{c\langle X\rangle}$ a certain-key-index for table I over (T, T_S). Define*

$$K := \{A \in X \mid t[A] \neq \perp\} .$$

Then the existence of a tuple in I weakly similar to t can be checked with 2^k lookups in $\mathfrak{I}_K$, where $k := |K \setminus T_S|$.

As $|K \setminus T_S|$ is bounded by $|X \setminus T_S|$, lookup is efficient whenever indexing is efficient.

Example 14 *Consider schema* $(ABCD, A, \{c\langle ABC\rangle\})$ *with table* I *over it:*

$$I = \begin{array}{|cccc|}\hline A & B & C & D \\ \hline 1 & \bot & \bot & 1 \\ 2 & 2 & \bot & 2 \\ 3 & \bot & 3 & \bot \\ 4 & 4 & 4 & \bot \\ \hline\end{array}$$

The certain-key-index $\mathfrak{I}_{c\langle ABC\rangle}$ *for* $c\langle ABC\rangle$ *consists of:*

$\mathfrak{I}_A$	$\mathfrak{I}_{AB}$		$\mathfrak{I}_{AC}$		$\mathfrak{I}_{ABC}$		
A	A	B	A	C	A	B	C
1	1	$\bot$	1	$\bot$	1	$\bot$	$\bot$
2	2	2	2	$\bot$	2	2	$\bot$
3	3	$\bot$	3	3	3	$\bot$	3
4	4	4	4	4	4	4	4

These tables represent attribute values that we index by, not tables that are actually stored. When checking whether tuple $t := (2, \bot, 3, 4)$ *is weakly similar on* ABC *to some tuple* $t' \in I$ *(and thus violating* $c\langle ABC\rangle$ *when inserted), we perform lookup on* $\mathfrak{I}_{AC}$ *for tuples* t' *with* $t'[AC] \in \{(2,3), (2,\bot)\}$.

We briefly comment on prefix indexing in Section C.

7 Experiments

We conducted several experiments to evaluate various aspects of our work. Firstly, we mined publicly available databases for possible and certain keys. Secondly, we tested our algorithms for computing and deciding the existence of Armstrong tables. Lastly, we considered storage space and time requirements for our index scheme. We used the following data sets for our experiments: i) GO-termdb (Gene Ontology) at `geneontology.org/`, ii) IPI (International Protein Index) at `ebi.ac.uk/IPI`, iii) LMRP (Local Medical Review Policy) from `cms.gov/medicare-coverage-database/`, iv) PFAM (protein families) at `pfam.sanger.ac.uk/`, and v) RFAM (RNA families) at `rfam.sanger.ac.uk/`. These were chosen for their availability in database format. For ease of testing, we excluded tables of size larger than 100MB.

7.1 Key Mining

Examining the schema definition does not suffice to decide what types of key constraints hold or should hold on a database. Certain keys with $\bot$ occurrences cannot be expressed in current SQL databases, so would be lost. Even constraints that could and should be expressed are often not declared. Furthermore, even if `NOT NULL` constraints are declared, one frequently finds that these are invalid, resulting from work-arounds such as empty strings. We therefore mined data tables for possible and certain key constraints, with

focus on finding certain keys containing $\perp$ occurrences. In order to decide whether a column is `NOT NULL`, we ignored any schema-level declarations, and instead tested whether a column contained $\perp$ occurrences. To alleviate string formatting issues (such as "Zwieb C." vs "Zwieb C.;") we normalized strings by trimming non-word non-decimal characters, and interpreting the empty string as $\perp$. This pre-processing was necessary because several of our test data sets were available only in CSV format, where $\perp$ occurrences would have been exported as empty strings. Tables containing less than two tuples were ignored. In the figures reported, we exclude tables `pfamA` and `lcd` from the PFAM and LMRP data sets, as they contain over 100,000 (`pfamA`) and over 2000 (`lcd`) minimal keys, respectively, almost all of which appear to be 'by chance', and thus would distort results completely. Table 2 lists the number of minimal keys of each type discovered in the 130 tables. We distinguish between possible and certain keys with `NULL` columns, and keys not containing `NULL` columns. For the latter, possible and certain keys coincide.

Key Type	Occurrences
Certain Keys with `NULL`	43
Possible Keys with `NULL`	237
Keys without `NULL`	87

Table 2: Keys found by type

Two factors are likely to have a significant impact on the figures above. First, constraints may only hold accidentally, especially when the tables examined are small. For example, Table 1 of the RFAM data set satisfies $c\,\langle title, journal\rangle$ and $c\,\langle author, journal\rangle$, with `NULL` column *journal*. Only the first key appears to be sensible. Second, constraints that should sensibly hold may well be violated due to lack of enforcement. Certain keys with $\perp$ occurrences, which cannot easily be expressed in existing systems, are likely to suffer from this effect more than others. We thus consider our results qualitative rather than quantitative. Still, they indicate that certain keys do appear in practice, and may benefit from explicit support by a database system.

7.2 Armstrong Tables

We applied Algorithm 1 and Construction 1 to compute Armstrong tables for the 130 tables we mined possible and certain keys for. As all certain keys contained some `NOT NULL` column, Corollary 3 applied in all cases, and each Armstrong table was computed within a few milliseconds. Each Armstrong table contained only 7 tuples on average.

We also tested our algorithms against 1 million randomly generated table schemas and sets of keys. Each table contained 5-25 columns, with each column having a 50% chance of being `NOT NULL`, and 5-25 keys of size 1-5, with equal chance of being possible or certain. To avoid overly silly examples, we removed certain keys with only 1 or 2 `NULL` attributes. Hence, case ii) of Corollary 3 never applies.

We found that in all but 85 cases, an Armstrong table existed, with average computation time again in the order of a few milliseconds. However, note that for larger random schemas, transversal computation can become a bottleneck.

Together these results support our intuition that although Algorithm 1 has exponential complexity in the worst case, such cases need to be carefully constructed and arise neither in practice nor by chance (at least not frequently).

7.3 Indexing

The efficiency of our index scheme for certain keys with $\perp$ occurrences depends directly on the number of `NULL` attributes in the certain key. Thus the central question becomes how many `NULL` attributes occur in certain keys in practice. For the data sets in our experiments, having 43 certain keys with $\perp$ occurrences, the distribution of the number of these occurrences is listed in Table 3.

#NULLs	Frequency
1	39
2	4

Table 3: `NULL` columns in certain keys

Therefore, certain keys with $\perp$ occurrences contain mostly only a single attribute on which $\perp$ occurs (requiring 2 standard indices), and never more than two attributes (requiring 4 standard indices). Assuming these results generalize to other data sets, we conclude that certain key with $\perp$ occurrences can be enforced efficiently in practice.

We used triggers to enforce certain keys under different combinations of B-tree indexes, and compared these to the enforcement of the corresponding primary keys. The experiments were run in MySQL version 5.6 on a Dell Latitude E5530, Intel core i7, CPU 2.9GHz with 8GB RAM on a 64-bit operating system. For all experiments the schema was $(T = ABCDE, T_S = A)$ and the table I over T contained 100M tuples. In each experiment we inserted 10,000 times one tuple and took the average time to perform this operation. This includes the time for maintaining the index structures involved. We enforced $c\langle X\rangle$ in the first experiment for $X = AB$, in the second experiment for $X = ABC$ and in the third experiment for $X = ABCD$, incrementing the number of `NULL` columns from 1 to 3. The distribution of permitted $\perp$ occurrences was evenly spread amongst the 100M tuples, and also amongst the 10,000 tuples to be inserted. Altogether, we run each of the experiments for 3 index structures: i) $\mathfrak{I}_X$, ii) $\mathfrak{I}_{T_S}$, iii) $\mathfrak{I}_{c\langle X\rangle}$. The times were compared against those achieved under declaring a primary key $PK(X)$ on X, where we had 100M X-total tuples. Our results are shown in Table 4. All times are in milliseconds.

	$T = ABCDE, T_S = A$		
Index	$X = AB$	$X = ABC$	$X = ABCD$
$PK(X)$	0.451	0.491	0.615
$\mathfrak{I}_X$	0.764	0.896	0.977
$\mathfrak{I}_{T_S}$	0.723	0.834	0.869
$\mathfrak{I}_{c\langle X\rangle}$	0.617	0.719	1.143

Table 4: Average Times to Enforce Keys on X

Hence, certain keys can be enforced efficiently as long as we involve the columns in T_S in some index. Just having $\mathfrak{I}_{T_S}$ ensures a performance similar to that of the corresponding primary key. Indeed, the `NOT NULL` attributes of a certain key suffice to identify most tuples uniquely. Our experiments confirm these observations even for certain keys with three `NULL` columns, which occur rarely in practice. Of course, $\mathfrak{I}_{T_S}$ cannot guarantee the efficiency bounds established for $\mathfrak{I}_{c\langle X\rangle}$ in Theorem 12. We stress the importance of indexing: enforcing $c\langle X\rangle$ on our data set without an index resulted in a performance loss in the order of 10^4.

8 Conclusion and Future Work

We studied keys over SQL tables in which null marker occurrences follow Codd's interpretation as "value exists but unknown". Naturally, this interpretation equips keys with a possible world semantics. Indeed, a key is possible (certain) to hold on a table if some (every) possible world of the table satisfies the key. Possible keys capture SQL's `unique` constraint, and certain keys generalize SQL's primary keys by permitting null markers to occur in key columns. We have established solutions to several computational problems related to possible and certain keys under `NOT NULL` constraints. These include axiomatic and linear-time algorithmic characterizations of the associated implication problem, minimal representations of keys, discovery of keys from a given table, structural and computational properties of Armstrong tables, as well as extremal set problems for keys. Experiments confirm that our solutions work efficiently in practice. This also applies to enforcing certain keys, by combining known index schemes. Indeed, our findings from public data confirm the intuition that certain keys have only few columns with null marker occurrences. In conclusion, certain keys achieve the goal of Codd's rule of entity integrity and allow the entry of any tuples that can be uniquely identified. This gives them a distinct advantage over primary keys, with just a minor trade-off in update performance.

Several open problems should be addressed in the future. The exact complexity of deciding the existence of Armstrong tables should be determined. It is worth investigating optimizations to reduce the time complexity of computing Armstrong tables, and their size. The actual usefulness of Armstrong tables for the acquisition of possible and certain keys should be investigated empirically, similar to classical functional dependencies [30]. Evidently, the existence and computation of Armstrong relations for sets of weak and strong functional dependencies requires new attention [33]. Approximate and scalable discovery is an important problem as meaningful possible or certain keys may be violated. This line of research has only been started yet for total [41] and possible keys [26]. Other index schemes may prove valuable to enforce certain keys. Finally, the possible worlds approach should be applied to other popular classes of constraints, including multivalued and inclusion dependencies.

Acknowledgement. We thank Georg Gottlob for comments on a draft. This research is supported by the Marsden Fund Council from New Zealand Government funding.

References

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[2] S. Abiteboul and V. Vianu. Transactions and integrity constraints. In *PODS*, pages 193–204, 1985.

[3] W. W. Armstrong. Dependency structures of data base relationships. In *IFIP Congress*, pages 580–583, 1974.

[4] P. Atzeni and N. M. Morfuni. Functional dependencies and constraints on null values in database relations. *Information and Control*, 70(1):1–31, 1986.

[5] C. Beeri, M. Dowd, R. Fagin, and R. Statman. On the structure of Armstrong relations for functional dependencies. *J. ACM*, 31(1):30–46, 1984.

[6] J. Biskup. *Security in Computing Systems - Challenges, Approaches and Solutions*. Springer, 2009.

[7] P. Buneman, S. B. Davidson, W. Fan, C. S. Hara, and W. C. Tan. Keys for XML. *Computer Networks*, 39(5):473–487, 2002.

[8] A. Calì, D. Calvanese, and M. Lenzerini. Data integration under integrity constraints. In *Seminal Contributions to Information Systems Engineering*, pages 335–352. 2013.

[9] E. F. Codd. Extending the database relational model to capture more meaning. *ACM Trans. Database Syst.*, 4(4):397–434, 1979.

[10] E. F. Codd. *The Relational Model for Database Management, Version 2*. Addison-Wesley, 1990.

[11] C. David, L. Libkin, and T. Tan. Efficient reasoning about data trees via integer linear programming. *ACM Trans. Database Syst.*, 37(3):19, 2012.

[12] A. Deutsch, L. Popa, and V. Tannen. Query reformulation with constraints. *SIGMOD Record*, 35(1):65–73, 2006.

[13] J. Diederich and J. Milton. New methods and fast algorithms for database normalization. *ACM Trans. Database Syst.*, 13(3):339–365, 1988.

[14] T. Eiter and G. Gottlob. Identifying the minimal transversals of a hypergraph and related problems. *SIAM J. Comput.*, 24(6):1278–1304, 1995.

[15] K. Engel. *Sperner Theory*. Cambridge Uni Press, 1997.

[16] R. Fagin. A normal form for relational databases that is based on domains and keys. *ACM Trans. Database Syst.*, 6(3):387–415, 1981.

[17] R. Fagin. Horn clauses and database dependencies. *J. ACM*, 29(4):952–985, 1982.

[18] R. Fagin, P. G. Kolaitis, R. J. Miller, and L. Popa. Data exchange: semantics and query answering. *Theor. Comput. Sci.*, 336(1):89–124, 2005.

[19] R. Fagin and M. Y. Vardi. The theory of data dependencies - an overview. In *ICALP*, pages 1–22, 1984.

[20] W. Fan, F. Geerts, and X. Jia. A revival of constraints for data cleaning. *PVLDB*, 1(2):1522–1523, 2008.

[21] F. Ferrarotti, S. Hartmann, V. Le, and S. Link. Codd table representations under weak possible world semantics. In *DEXA (1)*, pages 125–139, 2011.

[22] S. Hartmann, M. Kirchberg, and S. Link. Design by example for SQL table definitions with functional dependencies. *VLDB J.*, 21(1):121–144, 2012.

[23] S. Hartmann, U. Leck, and S. Link. On Codd families of keys over incomplete relations. *Comput. J.*, 54(7):1166–1180, 2011.

[24] S. Hartmann and S. Link. Efficient reasoning about a robust XML key fragment. *ACM Trans. Database Syst.*, 34(2), 2009.

[25] S. Hartmann and S. Link. The implication problem of data dependencies over SQL table definitions. *ACM Trans. Database Syst.*, 37(2):13, 2012.

[26] A. Heise, Jorge-Arnulfo, Quiane-Ruiz, Z. Abedjan, A. Jentzsch, and F. Naumann. Scalable discovery of unique column combinations. *PVLDB*, 7(4):301–312, 2013.

[27] A. K. Jha, V. Rastogi, and D. Suciu. Query evaluation with soft-key constraints. In *PODS*, pages 119–128, 2008.

[28] G. O. H. Katona. Combinatorial and algebraic results for database relations. In *ICDT*, pages 1–20, 1992.

[29] A. Klug and R. Price. Determining view dependencies using tableaux. *ACM Trans. Database Syst.*, 7(3):361–380, 1982.

[30] W.-D. Langeveldt and S. Link. Empirical evidence for the usefulness of Armstrong relations in the acquisition of meaningful FDs. *Inf. Syst.*, 35(3):352–374, 2010.

[31] V. Le, S. Link, and M. Memari. Discovery of keys from SQL tables. In *DASFAA (1)*, pages 48–62, 2012.

[32] V. Le, S. Link, and M. Memari. Schema- and data-driven discovery of SQL keys. *JCSE*, 6(3):193–206, 2012.

[33] M. Levene and G. Loizou. Axiomatisation of functional dependencies in incomplete relations. *Theor. Comput. Sci.*, 206(1-2):283–300, 1998.

[34] M. Levene and G. Loizou. A generalisation of entity and referential integrity. *ITA*, 35(2):113–127, 2001.

[35] Y. E. Lien. On the equivalence of database models. *J. ACM*, 29(2):333–362, 1982.

[36] H. Mannila and K.-J. Räihä. *Design of Relational Databases*. Addison-Wesley, 1992.

[37] B. Marnette, G. Mecca, and P. Papotti. Scalable data exchange with FDs. *PVLDB*, 3(1):105–116, 2010.

[38] J. Melton. ISO/IEC 9075-2: 2003 (SQL/foundation). ISO standard, 2003.

[39] J. Paredaens. What about constraints in RDF? In *Conceptual Modelling*, LNCS, pages 7–18, 2012.

[40] T. J. Schaefer. The complexity of satisfiability problems. In *STOC*, pages 216–226, 1978.

[41] Y. Sismanis, P. Brown, P. J. Haas, and B. Reinwald. GORDIAN: Efficient and scalable discovery of composite keys. In *VLDB*, pages 691–702, 2006.

[42] E. Sperner. Ein Satz über Untermengen einer endlichen Menge. *Math. Z.*, 27:544–548, 1928.

[43] B. Thalheim. On semantic issues connected with keys in relational databases permitting null values. *Elektr. Informationsverarb. Kybern.*, 25(1/2):11–20, 1989.

[44] D. Toman and G. E. Weddell. On keys and functional dependencies as first-class citizens in description logics. *J. Autom. Reasoning*, 40(2-3):117–132, 2008.

[45] C. Zaniolo. Database relations with null values. *J. Comput. System Sci.*, 28(1):142–166, 1984.

A Proofs

A.1 Section 3

Theorem 13 (Theorem 1 restated)
$X \subseteq T$ is a possible (certain) key for I iff no distinct tuples in I are strongly (weakly) similar on X.

Proof *(possible key $\Rightarrow$ not strongly similar)* Let X be a possible key for I and $t, t' \in I$ be distinct. Then there exists a possible word $\rho(I)$ of I for which X is a key. Denote by $\rho(t), \rho(t')$ the copies of t, t' in $\rho(I)$. Since X is a key for $\rho(I)$, there exists $A \in X$ with $\rho(t)[A] \neq \rho(t')[A]$. Since only $\perp$ values get replaced, this means that $t[A] \neq t'[A]$ or $t[A] = \perp$ or $t'[A] = \perp$ holds. In either case t, t' are not strongly similar.

(possible key $\Leftarrow$ not strongly similar) Let no two distinct tuples in I be strongly similar on X. We construct a possible world $\rho(I)$ of I by replacing every $\perp$ value occurring in I with a distinct domain value that did not occur in I previously[2]. Let again $\rho(t), \rho(t') \in \rho(I)$ be two distinct copies of $t, t' \in I$. Since t, t' are not strongly similar on X, there exists $A \in X$ with $t[A] \neq t'[A]$ or $t[A] = \perp$ or $t'[A] = \perp$. As we have chosen our replacement values to be unique in $\rho(I)$, we have $\rho(t)[A] \neq \rho(t')[A]$ in all three cases. Thus $\rho(t), \rho(t')$ are not similar on X, so X is a possible key for I.

(not certain key $\Rightarrow$ weakly similar) Let X not be a certain key for I. Then there exists a possible world $\rho(I)$ of I and distinct tuples $t, t' \in I$ such that $\rho(t)[X] \sim \rho(t')[X]$. Thus for every $A \in X$ we have $\rho(t)[A] = \rho(t')[A]$ and hence $t[A] = t'[A]$ or $t[A] = \perp$ or $t'[A] = \perp$, i.e., $t[X] \sim_w t'[X]$.

(not certain key $\Leftarrow$ weakly similar) Let $t, t' \in I$ with $t[X] \sim_w t'[X]$. Then we can construct a possible world $\rho(I)$ of I which replaces $\perp$ values on t, t' as follows:

[2]Such a replacement exists since domains are infinite and tables finite.

- If $t[A] = t'[A] = \bot$ let $\rho(t)[A] = \rho(t')[A]$ be arbitrary.
- If $t[A] = \bot \wedge t'[A] \neq \bot$ let $\rho(t)[A] = t'[A]$.
- If $t[A] \neq \bot \wedge t'[A] = \bot$ let $\rho(t')[A] = t[A]$.

In each case we have $\rho(t)[A] = \rho(t')[A]$ and hence $\rho(t)[X] \sim \rho(t')[X]$. Thus X is not a certain key for I.

Theorem 14 (Theorem 2 restated)
Let Σ be a set of possible and certain keys.

i) Σ implies $c\langle X\rangle$ iff $c\langle Y\rangle \in \Sigma$ for some $Y \subseteq X$ or $p\langle Z\rangle \in \Sigma$ for some $Z \subseteq X \cap T_S$.

ii) Σ implies $p\langle X\rangle$ iff $c\langle Y\rangle \in \Sigma$ or $p\langle Y\rangle \in \Sigma$ for some $Y \subseteq X$.

Proof The "if" directions follow from Theorem 1. We show the "only if" direction next.

i) Let $c\langle Y\rangle \notin \Sigma$ for every $Y \subseteq X$ and $p\langle Z\rangle \notin \Sigma$ for every $Z \subseteq X \cap T_S$. Consider a table $I = t, t'$ on (T, T_S) with the following properties:

   - $t = (0, \ldots, 0)$
   - $t'[X \cap T_S] = (0, \ldots, 0)$
   - $t'[X \setminus T_S] = (\bot, \ldots, \bot)$
   - $t'[T \setminus X] = (1, \ldots, 1)$

   Now consider Theorem 1. Since t, t' weekly agree on X only, the only certain keys $c\langle Y\rangle$ violated by I are those with $Y \subseteq X$. Hence no certain keys in Σ are violated by I. Since t, t' strongly agree on $X \cap T_S$ only, the only possible keys $p\langle Z\rangle$ violated by I are those with $Z \subseteq X \cap T_S$. Hence no possible keys in Σ are violated by I. Hence I respects Σ but violates $c\langle X\rangle$, so Σ does not imply $c\langle X\rangle$.

ii) Analogous with $t'[X] = (0, \ldots, 0)$.

This concludes the proof.

Theorem 15 (Theorem 3 restated)
Let Π be a set of strong and weak anti-keys.

i) Π implies $\neg_p\langle X\rangle$ iff $\neg_p\langle Y\rangle \in \Pi$ for some $Y \supseteq X$, or $\neg_c\langle Z\rangle \in \Pi$ for some Z with $X \subseteq Z \cap T_S$.

ii) Π implies $\neg_c\langle X\rangle$ iff $\neg_c\langle Y\rangle \in \Pi$ or $\neg_p\langle Y\rangle \in \Pi$ for some $Y \supseteq X$.

Proof The "if" directions follow from Theorem 1. We show the "only if" direction next.

i) Consider a table I on (T, T_S) constructed by adding tuples $t^s_Y, t^{s\prime}_Y$ for every strong anti-key $Y \in \Pi$, and tuples $t^w_Y, t^{w\prime}_Y$ for every weak anti-key $Y \in \Pi$, with the following properties:

- $t^s_Y[Y] = t^{s'}_Y[Y] = (i, \ldots, i)$
- $t^s_Y[T \setminus Y] = (j, \ldots, j)$
- $t^{s'}_Y[T \setminus Y] = (k, \ldots, k)$

- $t^w_Y[Y \cap T_S] = t^{w'}_Y[Y \cap T_S] = (i, \ldots, i)$
- $t^w_Y[Y \setminus T_S] = t^{w'}_Y[Y \setminus T_S] = (\bot, \ldots, \bot)$
- $t^w_Y[T \setminus Y] = (j, \ldots, j)$
- $t^{w'}_Y[T \setminus Y] = (k, \ldots, k)$

for distinct $i, j, k \neq \bot$. It is clear by construction and Theorem 1 that Π holds on I. If $\neg_p\langle Y\rangle \notin \Pi$ for every $Y \supseteq X$ and $\neg_c\langle Z\rangle \notin \Pi$ for every Z with $X \subseteq Z \cap T_S$, then for every tuple $t \in I$ there exists $A \in X$ such that $t[A]$ is unique or $\bot$. Hence no tuples in I are strongly similar on X, so $p\,\langle X\rangle$ holds on I.

ii) Modify the construction in i) by setting

- $t^w_Y[Y \setminus T_S] = t^{w'}_Y[Y \setminus T_S] = (i, \ldots, i)$

Again it is clear that Π holds on I. If Π contains no anti-key $Y \supseteq X$, then for every tuple $t \in I$ there exists $A \in X$ such that $t[A]$ is unique. Hence no tuples in I are weakly similar on X, so $c\,\langle X\rangle$ holds on I.

This concludes the proof.

Theorem 16 (Theorem 4 restated)
The following axioms are sound and complete for implication of possible and certain keys.

$$\textit{p-Extension:}\quad \frac{p\,\langle X\rangle}{p\,\langle XY\rangle} \qquad \textit{c-Extension:}\quad \frac{c\,\langle X\rangle}{c\,\langle XY\rangle}$$

$$\textit{Weakening:}\quad \frac{c\,\langle X\rangle}{p\,\langle X\rangle} \qquad \textit{Strengthening:}\quad \frac{p\,\langle X\rangle}{c\,\langle X\rangle}\, X \subseteq T_S$$

Proof Soundness follows from Theorem 1.

For showing completeness let $\Sigma \vDash c\,\langle X\rangle$. By Theorem 2 there exist either $c\,\langle Y\rangle \in \Sigma$ with $Y \subseteq X$ or $p\,\langle Z\rangle \in \Sigma$ with $Z \subseteq X \cap T_S$. In the former case $c\,\langle X\rangle$ can be derived via c-Extension, in the latter via Strengthening and c-Extension. Finally let $\Sigma \vDash p\,\langle X\rangle$. By Theorem 2 there exist $Y \subseteq X$ with $c\,\langle Y\rangle \in \Sigma$ or $p\,\langle Y\rangle \in \Sigma$. In the former case $p\,\langle X\rangle$ can be derived via Weakening and p-Extension, in the latter via p-Extension alone.

Lemma 6 (Lemma 1 restated)
The set Σ_{min} of all minimal possible and certain keys w.r.t. Σ is a non-redundant cover of Σ.

Proof For every key $\sigma \in \Sigma$ there exists a minimal key σ' with $\Sigma \vDash \sigma' \vDash \sigma$ (found through repeated application of Definition 4). Hence Σ_{min} is a cover of Σ.

Now assume $\sigma \in \Sigma_{min}$ is redundant. Then there must exist $\sigma' \in \Sigma_{min} \setminus \{\sigma\}$ with $\sigma' \vDash \sigma$. Since σ is minimal this can only happen for $\sigma = c\langle X\rangle, \sigma' = p\langle X\rangle$ for some $X \subseteq T_S$. But then σ' is not minimal, which contradicts $\sigma' \in \Sigma_{min}$. Hence Σ_{min} is non-redundant.

Theorem 17 (Theorem 5 restated)
Let I be a table over (T, T_S), and Σ_I the set of all certain and/or possible keys that hold on I. Then

$$\Sigma := \{c\langle X\rangle \mid X \in Tr(\overline{\mathcal{AG}^w})\} \cup \{p\langle X\rangle \mid X \in Tr(\overline{\mathcal{AG}^s})\}$$

is a cover of Σ_I.

Proof By Theorem 1 X is a certain (possible) key for I

iff no distinct tuples $t.t' \in I$ weakly (strongly) agree on X

iff X is not a subset of any weak (strong) agree set

iff X intersects with $\overline{Y}$ for every weak (strong) agree set Y

iff X is a transversal of $\overline{\mathcal{AG}^w}$ ($\overline{\mathcal{AG}^s}$)

Due to the extension rules of Theorem 4 the minimal transversals are sufficient to form a cover.

A.2 Section 4

Lemma 7 (Lemma 2 restated)
Let I be an instance over (T, T_S) with $|I| \geq 2$. Then I violates $c\langle \perp\text{-base}(t, t')\rangle$ for every $t, t' \in I$.

Proof Let $X := \perp\text{-base}(t, t')$. If $t \neq t'$ then $t[X] \sim_w t'[X]$ so $c\langle X\rangle$ is violated. Otherwise $t = t'$ and $t[X] = (\perp, \ldots, \perp)$. Since I contains at least two tuples, there must exist $t'' \in I$ with $t \neq t''$, and we have $t[X] \sim_w t''[X]$ so $c\langle X\rangle$ is again violated.

Theorem 18 (Theorem 6 restated)
Let I be an instance over (T, T_S) such that Σ holds on I. Then I is a pre-Armstrong table of Σ iff

i) for every strong anti-key $\neg_p\langle X\rangle \in \mathcal{A}^s_{max}$ there exist distinct tuples $t, t' \in I$ with $t[X] \sim_s t'[X]$, and

ii) for every weak anti-key $\neg_c\langle X\rangle \in \mathcal{A}^w_{max}$ there exist distinct tuples $t, t' \in I$ with $t[X] \sim_w t'[X]$.

Proof Keys implied by Σ hold on I by assumption, so we only need to examine keys on T that are not implied.

($\Rightarrow$) Clear by Theorem 1.

($\Leftarrow$) Assume *i)* and *ii)* hold. Let $c\langle X\rangle, X \subseteq T$ be a certain key with $\Sigma \not\models c\langle X\rangle$. Then $\neg_c\langle X\rangle$ is a weak anti-key and there exists a maximal (weak or strong) anti-key $\neg\langle Y\rangle$ with $X \subseteq Y$. In either case (by *i)* or *ii)*) there exist $t, t' \in I, t \neq t'$ with $t[Y] \sim_w t'[Y]$ and hence $t[X] \sim_w t'[X]$. Thus $c\langle X\rangle$ is violated by I.

Let $p\langle X\rangle, X \subseteq T$ be a possible key with $\Sigma \not\models p\langle X\rangle$. Then $\neg_p\langle X\rangle$ is a strong anti-key and there exists a maximal strong anti-key $\neg_p\langle Y\rangle$ with $X \subseteq Y$. By *ii)* there exist $t, t' \in I, t \neq t'$ with $t[Y] \sim_s t'[Y]$ and hence $t[X] \sim_s t'[X]$. Thus $p\langle X\rangle$ is violated by I.

This concludes the proof.

Theorem 19 (Theorem 7 restated)
Let $\Sigma \not\models c\langle\emptyset\rangle$. There exists a pre-Armstrong table for (T, T_S, Σ) iff there exists a set $\mathcal{W} \subseteq \mathcal{P}(T \setminus T_S)$ with the following properties:

i) Every element of $\mathcal{W}^{\times 2}$ forms a weak anti-key.

ii) For every maximal weak anti-key $\neg_c\langle X\rangle \in \mathcal{A}^w_{max}$ there exists $Y \in \mathcal{W}^{\times 2}$ with $Y \cap X' \neq \emptyset$ for every possible key $p\langle X'\rangle \in \Sigma$ with $X' \subseteq X$.

There exists an Armstrong table for (T, T_S, Σ) iff i) and ii) hold as well as

iii) $\bigcup \mathcal{W} = T \setminus T_S$.

Proof Let I be a pre-Armstrong table for (T, T_S, Σ) and $\Sigma \not\models c\langle\emptyset\rangle$. Define $\mathcal{W} \subseteq \mathcal{P}(T \setminus T_S)$ as

$$\mathcal{W} := \{\bot\text{-base}(t) \mid t \in I\}$$

We will show that conditions *i)* and *ii)* hold.

i) Let $Y_1, Y_2 \in \mathcal{W}$. Then there must exist $t_{Y_1}, t_{Y_2} \in I$ with $\bot\text{-base}(t_{Y_1}, t_{Y_2}) = Y_1Y_2$, so I violates $c\langle Y_1Y_2\rangle$ by Lemma 2. Since I respects Σ, Σ permits $\neg_c\langle Y_1Y_2\rangle$.

ii) For every maximal weak anti-key $\neg_c\langle X\rangle \in \mathcal{A}^w_{max}$ there exist two distinct tuples $t, t' \in I$ such that $t[X] \sim_w t'[X]$. Now let $p\langle X'\rangle \in \Sigma$ with $X' \subseteq X$. Since I respects Σ we cannot have $t[X'] \sim_s t'[X']$, i.e., t, t' are weakly but not strongly similar on X'. This is only possible for $t[A] = \bot$ or $t'[A] = \bot$ for some $A \in X'$. But that means $A \in X' \cap Y_1Y_2$ with $Y_1 := \bot\text{-base}(t), Y_2 := \bot\text{-base}(t')$.

If I is an Armstrong table, then there exists a tuple $t_A \in I$ with $t_A[A] = \bot$ for every $A \in T \setminus T_S$. Hence

iii)
$$T \setminus T_S \subseteq \bigcup\{\bot\text{-base}(t_A) \mid A \in T \setminus T_S\} \subseteq \bigcup \mathcal{W}$$

The "if" direction will be shown in Theorem 8.

Theorem 20 (Theorem 8 restated)
Let $\Sigma \not\models c\langle\emptyset\rangle$ and I constructed via Construction 1. Then I is a pre-Armstrong table over (T, T_S, Σ). If condition iii) of Theorem 7 holds for $\mathcal{W}$, then I is an Armstrong table.

Proof We begin by showing that the conditions of Theorem 6 are met.

i) Let $\neg_p\langle X\rangle$ be a strong anti-key in $\mathcal{A}^s_{max}$. Then I contains the tuples $t^s_X, t^{s\prime}_X$ with

$$t^s_X[X] = (i,\ldots,i) \sim_s (i,\ldots,i) = t^{s\prime}_X[X]$$

ii) Let $\neg_c\langle X\rangle$ be a weak anti-key in $\mathcal{A}^w_{max}$. Then I contains the tuples $t^w_X, t^{w\prime}_X$ with

$$\begin{aligned} t^w_X[X \setminus Y_1Y_2] = (i,\ldots,i) &\sim_w (i,\ldots,i) = t^{w\prime}_X[X \setminus Y_1Y_2] \\ t^w_X[X \cap Y_1] = (\bot,\ldots,\bot) &\sim_w t^{w\prime}_X[X \cap Y_1] \\ t^w_X[X \cap Y_2] &\sim_w (\bot,\ldots,\bot) = t^{w\prime}_X[X \cap Y_2] \end{aligned}$$

It remains to show that I is a valid instance over (T, T_S, Σ), i.e., that I honors T_S and does not violate constrains in Σ. Honoring of T_S is clear by choice of $Y, Z, A \subseteq T \setminus T_S$.

i) Let $p\,\langle X'\rangle \in \Sigma$ and $t, t' \in I, t \neq t'$ with $t[X'] \sim_s t'[X']$. Since $X' \neq \emptyset$ and tuples constructed for different anti-keys use unique values, we must have $\{t, t'\} = \{t^s_X, t^{s\prime}_X\}$ or $\{t, t'\} = \{t^w_X, t^{w\prime}_X\}$ for some maximal anti-key $\neg\,\langle X\rangle$ with $X' \subseteq X$. The former cannot happen since $p\,\langle X'\rangle \in \Sigma$ implies $\Sigma \vDash p\,\langle X\rangle$, so $\{t, t'\} = \{t^w_X, t^{w\prime}_X\}$ and $\neg_c\langle X\rangle$ is a maximal weak anti-key. But then $\bot$-base(t, t') intersects with X' due to condition *ii*), so $t[X'] \sim_s t'[X']$ cannot hold.

ii) Let $c\,\langle X'\rangle \in \Sigma$ and $t, t' \in I, t \neq t'$ with $t[X'] \sim_w t'[X']$.

- If $X' \not\subseteq \bot$-base(t, t') we again must have $\{t, t'\} = \{t^s_X, t^{s\prime}_X\}$ or $\{t, t'\} = \{t^w_X, t^{w\prime}_X\}$ for some maximal anti-key $\neg\,\langle X\rangle$ with $X' \subseteq X$. Either way $\neg_c\langle X\rangle$ and thus $\neg_c\langle X'\rangle$ is a weak anti-key, which contradicts $\Sigma \vDash c\,\langle X'\rangle$.
- If $X' \subseteq \bot$-base(t, t') then $X' \subseteq Y$ for some $Y \in \mathcal{W}^{\times 2}$. But $\neg_c\langle Y\rangle$ is a weak anti-key by condition *i*), again contradicting $\Sigma \vDash c\,\langle X'\rangle$.

If condition *iii*) of Theorem 7 holds for $\mathcal{W}$, then construction step *III*) ensures that I is an Armstrong table.

Corollary 4 (Corollary 3 restated)
Let $\Sigma \nvDash c\,\langle\emptyset\rangle$.

i) If $\Sigma \nvDash c\,\langle X\rangle$ for every $X \subseteq T \setminus T_S$ then there exists an Armstrong table for (T, T_S, Σ).

ii) If $\Sigma \vDash c\,\langle X\rangle$ for any $X \subseteq T \setminus T_S$ with $|X| \leq 2$ then there does not exist an Armstrong table for (T, T_S, Σ).

Proof i) follows from Theorem 7 with $\mathcal{W} = \{T \setminus T_S\}$.
For case ii), let I be an Armstrong table for (T, T_S, Σ), and $AB \subseteq T \setminus T_S$. Then I contains tuples t_A, t_B with $t_A[A] = \bot$ and $t_B[B] = \bot$, so $AB \subseteq \bot$-base(t_A, t_B) is a weak anti-key by Lemma 2.

Lemma 8
A schema $(T, T_S, \Sigma, \mathcal{A}^w)$ is satisfiable iff there exists a set $\mathcal{W} \subseteq \mathcal{P}(T \setminus T_S)$ with the following properties:

i) Every element of $\mathcal{W}^{\times 2}$ forms a weak anti-key w.r.t. Σ.

ii) For every weak anti-key $\neg_c\langle X\rangle \in \mathcal{A}^w$ there exists $Y \in \mathcal{W}^{\times 2}$ with $Y \cap X' \neq \emptyset$ for every possible key $p\langle X'\rangle \in \Sigma$ with $X' \subseteq X$.

Proof Analogous to the proof of Theorem 7.

Theorem 21 (Theorem 9 restated) *The key/anti-key satisfiability problem is NP-complete.*

Proof We will reduce the monotone 1-in-3 SAT problem to it. Let $\mathcal{S}^{AT}$ be any set of 3-clauses without negation. We construct an instance of the key/anti-key satisfiability problem as follows.

$$\begin{aligned} T &:= XYZ \cup \bigcup_{\mathcal{C} \in \mathcal{S}^{AT}} \mathcal{C} \\ \mathcal{A}^w &:= \{\neg_c\langle XABC\rangle, \neg_c\langle YABC\rangle \mid ABC \in \mathcal{S}^{AT}\} \cup \{Z\} \\ \Sigma &:= \{p\langle A\rangle \mid A \in T\} \cup \\ &\qquad \left\{ \begin{array}{l} c\langle ZABC\rangle, c\langle XYAB\rangle, \\ c\langle XYAC\rangle, c\langle XYBC\rangle \end{array} \;\middle|\; ABC \in \mathcal{S}^{AT} \right\} \end{aligned}$$

We claim that $(T, \emptyset, \Sigma, \mathcal{A}^w)$ is satisfiable iff $\mathcal{S}^{AT}$ is 1-in-3 satisfiable.

Let I be a table satisfying $(T, \emptyset, \Sigma, \mathcal{A}^w)$, and $\mathcal{W}$ as in Lemma 8. We may assume w.l.o.g. that $\mathcal{W}$ is *downward closed*, i.e., that

$$\mathcal{W} = \bigcup_{w \in \mathcal{W}} \mathcal{P}(w)$$

Since every attribute in T is nullable and a possible key, we must have $\mathcal{A}^w \subseteq \mathcal{W}^{\times 2}$. In particular $Z \in \mathcal{W}$, and for every $ABC \in \mathcal{S}^{AT}$ we have $XABC, YABC \in \mathcal{W}^{\times 2}$. Since $c\langle ZABC\rangle \in \Sigma$ we cannot have $ABC \in \mathcal{W}$. This leaves XA, XB or $XC \in \mathcal{W}$, and similarly YA, YB or $YC \in \mathcal{W}$. The certain keys $c\langle XYAB\rangle, c\langle XYAC\rangle, c\langle XYBC\rangle \in \Sigma$ mean that $XA \in \mathcal{W}$ prohibits both $YB \in \mathcal{W}$ and $YC \in \mathcal{W}$ so $YA \in \mathcal{W}$ must hold. Conversely $YA \in \mathcal{W}$ prohibits both $XB \in \mathcal{W}$ and $XC \in \mathcal{W}$. By symmetrical argument exactly one of XA, XB, XC lies in $\mathcal{W}$.

Thus $\mathcal{S}^{AT}$ is 1-in-3 satisfiable with

$$A \to \begin{cases} \text{true} & \text{if } XA \in \mathcal{W} \\ \text{false} & \text{if } XA \notin \mathcal{W} \end{cases}$$

Conversely let $\mathcal{S}^{AT}$ be 1-in-3 satisfiable with truth assignment $\mathcal{L}$. Then the set

$$\mathcal{W} := \{XA, YA, BC \mid ABC \in \mathcal{S}^{AT} \wedge \mathcal{L}(A)\} \cup \{Z\}$$

meets the conditions of Lemma 8.

The above shows NP-hardness. Furthermore, if there exists an instance I satisfying the given keys and anti-keys, it contains a subtable with no more than $2 \cdot |\mathcal{A}^w|$ tuples, and thus can be guessed and verified in polynomial time.

Lemma 9 (Lemma 3 restated)
Let Σ^P, Σ^C be the sets of possible and certain keys in Σ, and for the sake of readability we will identify attribute sets with the keys or anti-keys induced by them. Then

$$\begin{aligned}\mathcal{A}^s_{max} = \{\ X \in \overline{Tr(\Sigma^C \cup \Sigma^P)} \mid \\ \neg(X \subseteq T_S \wedge \exists Y \in \mathcal{A}^w_{max}.\ X \subset Y)\ \} \\ \mathcal{A}^w_{max} = \overline{Tr(\Sigma^C \cup \{X \in \Sigma^P \mid X \subseteq T_S\})} \setminus \mathcal{A}^s_{max}\end{aligned}$$

Proof $\neg\langle X\rangle$ is a strong (weak) anti-key iff $p\langle X\rangle$ ($c\langle X\rangle$) is not implied by Σ. By Theorem 2 $p\langle X\rangle$ ($c\langle X\rangle$) is implied by Σ iff there exists $Y \subseteq X$ with $Y \in \mathcal{K}^P := \Sigma^C \cup \Sigma^P$ ($Y \in \mathcal{K}^C := \Sigma^C \cup \{X \in \Sigma^P \mid X \subseteq T_S\}$). Hence $\neg\langle X\rangle$ is a strong (weak) anti-key iff it is not a subset of any set in $\mathcal{K}^P$ ($\mathcal{K}^C$), i.e., iff $\overline{X}$ intersects with every set in $\mathcal{K}^P$ ($\mathcal{K}^C$). But this holds iff $\overline{X}$ is a transversal set for $\mathcal{K}^P$ ($\mathcal{K}^C$). Thus the *set-wise maximal* strong (weak) anti-keys are exactly $\overline{Tr(\mathcal{K}^P)}$ ($\overline{Tr(\mathcal{K}^C)}$).

By Definition 3, a strong anti-key $\neg_p\langle X\rangle$ is maximal iff it is not implied by a strictly larger anti-key, i.e., iff $\neg_p\langle X\rangle$ is set-wise maximal and not implied by a strictly larger weak anti-key $\neg_c\langle Y\rangle$. By Theorem 3 such a weak anti-key $\neg_c\langle Y\rangle$ implies $\neg_p\langle X\rangle$ iff $X \subseteq T_S$.

Again by Definition 3, a weak anti-key is maximal iff it is set-wise maximal and not a strong anti-key. Furthermore, if it were set-wise maximal and a strong anti-key, it would have to be a maximal strong anti-key.

This gives use exactly the conditions of Lemma 3.

Lemma 10 (Lemma 4 restated)
Let $\mathcal{T}_X$ be the transversal set in $T \setminus T_S$ of all possible keys that are subsets of X, and $\mathcal{T}$ the set of all such transversals for all maximal weak antikeys:

$$\begin{aligned}\mathcal{T}_X &:= Tr(\{X' \cap (T \setminus T_S) \mid p\langle X'\rangle \in \Sigma_{min} \wedge X' \subseteq X\}) \\ \mathcal{T} &:= \{\mathcal{T}_X \mid X \in \mathcal{A}^w_{max}\}\end{aligned}$$

Then condition ii) of Theorem 7 can be rephrased as follows:

ii') $\mathcal{W}^{\times 2}$ $\wedge\vee$-supports $\mathcal{T}$.

Proof Condition ii) states that for every $\neg_c\langle X\rangle \in \mathcal{A}^w_{max}$ there exists $Y \in \mathcal{W}^{\times 2}$ which traverses $\{X' \mid p\langle X'\rangle \in \Sigma_{min} \wedge X' \subseteq X\}$. Since $Y \subseteq T \setminus T_S$ this means Y traverses $\{X' \cap (T \setminus T_S) \mid p\langle X'\rangle \in \Sigma_{min} \wedge X' \subseteq X\}$. Hence Y is a superset of a minimal transversal in $\mathcal{T}_X$, i.e. $\mathcal{W}^{\times 2}$ $\vee$-supports $\mathcal{T}_X$. This holds for every $\neg_c\langle X\rangle \in \mathcal{A}^w_{max}$, so $\mathcal{W}^{\times 2}$ $\wedge\vee$-supports $\mathcal{T}$. Each of these deductions is a logical equivalence.

Lemma 11 (Lemma 5 restated)
Algorithm Armstrong-Set is correct.

(Sketch) In lines 1 and 3 the conditions of Corollary 3 are checked, so any set $\mathcal{W}$ returned here meets conditions i) to iii).

Failing that, $\mathcal{W}$ is initialized so that condition iii) of Theorem 7 holds. Since function Extend-Support returns a superset of $\mathcal{W}$ condition iii) is an invariant for $\mathcal{W}$.

Condition i) of Theorem 7 holds for the initial $\mathcal{W}$ due to the check in line 3. Any subsequent enlargements of $\mathcal{W}$ in line 18 ensure that condition i) is maintained, due to the check in line 17. Hence condition i) is invariant for $\mathcal{W}$.

It remains to show condition ii). For $\mathcal{W} = \{\{A\} \mid A \in T \setminus T_S\}$ the set $\mathcal{W}^{\times 2}$ already $\vee$-supports transversal sets containing transversals of cardinality 2 or less. Thus for any extension $\mathcal{W}_+$ of $\mathcal{W}$ for which $\mathcal{W}^{\times 2}$ $\wedge\vee$-supports the set $\mathcal{T}$ after removal of such transversal sets in line 8, the set $\mathcal{W}_+^{\times 2}$ $\wedge\vee$-supports the original $\mathcal{T}$. Thus condition ii) follows directly from the correctness of function Extend-Support and Lemma 4. This correctness can be shown recursively: If the check in line 13 holds $\mathcal{W}^{\times 2}$ already $\vee$-supports $\mathcal{T}_X$. If it does not, $\vee$-support of $\mathcal{T}_X$ is ensured by extending $\mathcal{W}$ with Y_1, Y_2 in line 18. The recursive call in line 14 or 18 ensures that $\mathcal{W}'^{\times 2}$ $\wedge\vee$-supports $\mathcal{T} \setminus \{\mathcal{T}_X\}$.

It remains to argue that function Armstrong-Set returns such a set $\mathcal{W}$ (rather than $\perp$) whenever one exists. Essentially we are examining all minimal[3] sets $\mathcal{W}$ which meet conditions ii) and iii), with a shortcut in line 3 which is justified due to Corollary 3. In lines 8 and 14 we only omit cases leading to non-minimal sets $\mathcal{W}$. Since condition i) holds for a set $\mathcal{W}$ if it holds for any larger[3] set $\mathcal{W}'$, examining only minimal sets $\mathcal{W}$ is sufficient to find a set $\mathcal{W}$ meeting all conditions of Theorem 7, should one exist.

A.3 Section 5

Theorem 22 (Theorem 10 restated)
Σ is non-redundant if and only if the following conditions are satisfied

(1) $\mathcal{F}$ *is an antichain,*

(2) $\mathcal{G}$ *is an antichain,*

(3) $\forall\, F \in \mathcal{F}, G \in \mathcal{G} :\ F \not\subseteq G$,

(4) $\forall\, F \in \mathcal{F}, G \in \mathcal{G} :\ G \not\subseteq F \cap A$.

Proof Σ is non-redundant if and only if for every $\sigma \in \Sigma$ we have $\Sigma \setminus \{\sigma\} \not\models \sigma$. By Theorem 2, the latter condition is equivalent to saying that

(1') $\neg\exists c\,\langle X\rangle, c\,\langle Y\rangle \in \Sigma$ such that $Y \subseteq X$

(2') $\neg\exists p\,\langle X\rangle, p\,\langle Y\rangle \in \Sigma$ such that $Y \subseteq X$

(3') $\neg\exists p\,\langle X\rangle, c\,\langle Y\rangle \in \Sigma$ such that $Y \subseteq X$

(4') $\neg\exists c\,\langle X\rangle, p\,\langle Y\rangle \in \Sigma$ such that $Y \subseteq X \cap A$

hold. Conditions (1')–(4') are equivalent to conditions *(1)–(4)*.

The two lemmas below will be applied in the proof of Theorem 11. To prove the lemmas, we proceed similar to the original proof of Sperner's Theorem [42]. For $A \subsetneq [n]$

[3] w.r.t. the $\wedge$-support ordering $\mathcal{W} \lesssim \mathcal{W}' :\Leftrightarrow \forall Y \in \mathcal{W}.\exists Y' \in \mathcal{W}'.Y \subseteq Y'$

and $\mathcal{K} \subseteq [n+1]^{(i)}$, we will use the following notations:

$$\begin{aligned}
\Delta\mathcal{K} &:= \big\{X \in [n+1]^{(i-1)} : X \subset K \text{ for some } K \in \mathcal{K}\big\},\\
\tilde{\Delta}\mathcal{K} &:= \big\{X \in \Delta\mathcal{K} : n+1 \notin X \text{ or } X \not\subseteq A \cup \{n+1\}\big\},\\
\nabla\mathcal{K} &:= \big\{X \in [n+1]^{(i+1)} : X \supset K \text{ for some } K \in \mathcal{K}\big\},\\
\tilde{\nabla}\mathcal{K} &:= \big\{X \in \nabla\mathcal{K} : n+1 \notin X \text{ or } X \not\subseteq A \cup \{n+1\}\big\}.
\end{aligned}$$

The intuition here is that sets in $\mathcal{K}$ containing $n+1$ represent possible keys (after removing $n+1$), or elements of $\mathcal{G}$, while others represent certain keys, or elements of $\mathcal{F}$.

For $\emptyset \neq \mathcal{K} \subseteq [n+1]^{(i)}$, the normalized matching inequality [15, 42] says

$$\frac{|\Delta\mathcal{K}|}{|\mathcal{K}|} \geq \frac{\big|[n+1]^{(i-1)}\big|}{|[n+1]^{(i)}|} = \frac{i}{n+2-i}$$

or symmetrically,

$$\frac{|\nabla\mathcal{K}|}{|\mathcal{K}|} \geq \frac{\big|[n+1]^{(i+1)}\big|}{|[n+1]^{(i)}|} = \frac{n+1-i}{i+1}\,.$$

Lemma 12 *Let n, a, A be as in Theorem 11, and let $i \leq n+1$ be a nonnegative integer. Furthermore, let $\emptyset \neq \mathcal{K} \subseteq [n+1]^{(i)}$ such that there is no $K \in \mathcal{K}$ with $n+1 \in K \subseteq A \cup \{n+1\}$.*

(a) *If $i \leq n/2$, then $|\tilde{\nabla}\mathcal{K}| \geq |\mathcal{K}|$.*
For $n > 2$, equality is attained if and only if n is even, $i = n/2$, $a \leq n/2-2$ or $a = n-1$, and

$$\mathcal{K} = [n+1]^{(n/2)} \setminus \big(A^{(n/2-1)} \times \{n+1\}\big)$$

(b) *If $i \geq (n+3)/2$, then $|\tilde{\Delta}\mathcal{K}| > |\mathcal{K}|$.*

Proof (a) Assume that $i \leq n/2$. Consider the bipartite graph G on the vertex set $V(G) = \mathcal{K} \cup \tilde{\nabla}\mathcal{K}$ and with edge set

$$E(G) = \{(X, Y) : X \in \mathcal{K},\, Y \in \tilde{\nabla}\mathcal{K},\, X \subset Y\}.$$

Let $\mathcal{A} := \{X \in \mathcal{K} : X \subseteq A\}$. Now $X \in \mathcal{K}$ has degree $n-i$ in G if $X \in \mathcal{A}$ and degree $n-i+1$ otherwise. Hence,

$$|E(G)| = (n-i+1)(|\mathcal{K}| - |\mathcal{A}|) + (n-i)|\mathcal{A}| = (n-i+1)|\mathcal{K}| - |\mathcal{A}|. \tag{2}$$

As an immediate consequence of (2) we obtain

$$|E(G)| \geq (n-i)|\mathcal{K}|. \tag{3}$$

On the other hand, as every $Y \in \tilde{\nabla}\mathcal{K}$ is adjacent to at most $i+1$ elements of $\mathcal{K}$, we have

$$|E(G)| \leq (i+1)|\tilde{\nabla}\mathcal{K}|. \tag{4}$$

By (3) and (4), we have

$$(n-i)|\mathcal{K}| \leq (i+1)|\tilde{\nabla}\mathcal{K}|. \tag{5}$$

If $i < (n-1)/2$, then (5) implies $|\tilde{\nabla}\mathcal{K}| > |\mathcal{K}|$.

Assume that $i = (n-1)/2$, where n is odd. In this case, (5) reads $|\tilde{\nabla}\mathcal{K}| \geq |\mathcal{K}|$, and we need to show that equality can not hold. Assume that $|\tilde{\nabla}\mathcal{K}| = |\mathcal{K}|$. For this to be the case, equality must also hold in (3) and in (4). Equality in (3) means $\mathcal{K} = \mathcal{A}$. For equality in (4) it is necessary that every $Y \in \tilde{\nabla}\mathcal{K}$ is adjacent to exactly $(n+1)/2$ elements of $\mathcal{K}$, i.e., that $\Delta(\tilde{\nabla}\mathcal{K}) \subseteq \mathcal{K}$. Consider some $x \in K \in \mathcal{K}$ and $y \in [n] \setminus A$ (which exists as $a < n$). Then the set $K \cup \{y\} \setminus \{x\}$ is in $\Delta(\tilde{\nabla}\mathcal{K})$ but not in $\mathcal{K} = \mathcal{A}$, a contradiction.

For the remainder of the proof of (a), assume that $i = n/2$, where n is even. In this case, equation (2) becomes

$$|E(G)| = \left(\frac{n}{2}+1\right)|\mathcal{K}| - |\mathcal{A}|. \tag{6}$$

We will show the following strengthening of (4):

$$|E(G)| \leq \left(\frac{n}{2}+1\right)|\tilde{\nabla}\mathcal{K}| - |\mathcal{A}|. \tag{7}$$

Together with (6), this implies the claim $|\tilde{\nabla}\mathcal{K}| \geq |\mathcal{K}|$.

To prove (7), we will show that at least $|\mathcal{A}|$ elements of $\tilde{\nabla}\mathcal{K}$ have degree at most $n/2$ in G. As $a < n$, there is an $y \in [n] \setminus A$. Let $\mathcal{A}' := \mathcal{A} \times \{y\}$. Clearly, $\mathcal{A}'$ is a subset of $\tilde{\nabla}\mathcal{K}$, and $|\mathcal{A}'| = |\mathcal{A}|$. Consider the collection $\mathcal{B} \times \{y\}$ of those elements of $\mathcal{A}'$ that have degree $n/2+1$ in G. The remaining $|\mathcal{A}| - |\mathcal{B}|$ elements of $\mathcal{A}'$ have degree at most $n/2$ in G. By the choice of $\mathcal{B}$ we have $\Delta(\mathcal{B} \times \{y\}) \subseteq \mathcal{K}$ which implies

$$\Delta\mathcal{B} \times \{y, n+1\} \subseteq \tilde{\nabla}\mathcal{K} \setminus \mathcal{A}'. \tag{8}$$

As $\Delta\mathcal{B} \times \{n+1\}$ is a subset of $A^{(n/2-1)} \times \{n+1\}$, it does not contain any element of $\mathcal{K}$. That is, all $|\Delta\mathcal{B}|$ elements of $\Delta\mathcal{B} \times \{y, n+1\}$ have degree at most $n/2$ in G. In total, we have shown the number of elements of $\tilde{\nabla}\mathcal{K}$ that have degree at most $n/2$ in G to be at least $|\mathcal{A}| - |\mathcal{B}| + |\Delta\mathcal{B}|$. Since $\mathcal{B} \subseteq A^{(n/2)}$ and $a < n$, the normalized matching inequality gives $|\Delta\mathcal{B}| \geq |\mathcal{B}|$. This concludes the proof of (7).

It is easy to verify that $|\tilde{\nabla}\mathcal{K}| = |\mathcal{K}|$ holds for $\mathcal{K}$ as given in Lemma 12(a). It remains to show that for $n > 2$, $|\tilde{\nabla}\mathcal{K}| = |\mathcal{K}|$ implies that $\mathcal{K}$ is as in Lemma 12(a). Assume that $|\tilde{\nabla}\mathcal{K}| = |\mathcal{K}|$. Clearly, equality must hold in (6) and (7) then. By the above discussion, a necessary condition for equality in (7) is $|\Delta\mathcal{B}| = |\mathcal{B}|$.

Case 1. Assume that $\mathcal{B} = \emptyset$. Then, by the definition of $\mathcal{B}$ and for equality in (7), all elements of $\mathcal{A}'$ have degree $n/2$ in G, the remaining elements of $\tilde{\nabla}\mathcal{K}$ degree $n/2+1$. If $\mathcal{A} \neq \emptyset$, then consider some $Z \in \mathcal{A}$. The set $Z \cup \{y\}$ is in $\tilde{\nabla}\mathcal{K}$, and, as it has degree $n/2 > 1$ in G, there is a $z \in Z$ such that $Z \cup \{y\} \setminus \{z\} \in \mathcal{K}$. Now $Z' := Z \cup \{y, n+1\} \setminus \{z\}$ is in $\tilde{\nabla}\mathcal{K} \setminus \mathcal{A}'$. As $Z \cup \{n+1\} \setminus \{z\} \notin \mathcal{K}$, the set Z' has degree at most $n/2$ in G, a contradiction. Hence, $\mathcal{A} = \emptyset$. It follows that $\tilde{\nabla}\mathcal{K} = \nabla\mathcal{K}$. By the normalized matching inequality, we have $|\nabla\mathcal{K}| \geq |\mathcal{K}|$, and it is well-known (see [15], for instance) that equality only holds for $\mathcal{K} = [n+1]^{(n/2)}$, i.e., when $a \leq n/2 - 2$ and $\mathcal{K}$ as claimed.

Case 2. Assume that $\mathcal{B} \neq \emptyset$. Using the normalized matching inequality again, we obtain that $|\Delta\mathcal{B}| = |\mathcal{B}|$ is only possible when $a = n-1$ and $\mathcal{B} = \mathcal{A} = A^{(n/2)}$. Now (8) implies

$$\nabla\big(\mathcal{K} \,\dot\cup\, (A^{(n/2-1)} \times \{n+1\})\big) = \tilde\nabla\mathcal{K} \,\dot\cup\, (A^{(n/2)} \times \{n+1\}),$$

and the normalized matching inequality gives

$$|\mathcal{K}| + \binom{n-1}{n/2-1} \leq |\tilde\nabla\mathcal{K}| + \binom{n-1}{n/2}$$

(which is equivalent to $|\mathcal{K}| \leq |\tilde\nabla\mathcal{K}|$), where equality holds if and only if

$$\mathcal{K} = [n+1]^{(n/2)} \setminus \big(A^{(n/2-1)} \times \{n+1\}\big).$$

(b) The proof of Lemma 12(b) is analogous to the proof of Lemma 12(a) for $i \leq (n-1)/2$.

Lemma 13 *Let n, a, k, A, m be as in Theorem 11. Furthermore, let $\mathcal{H} \subseteq [n+1]^{(\leq k)}$ be an antichain such that there is no $H \in \mathcal{H}$ with $n+1 \in H \subseteq A \cup \{n+1\}$. Then*

$$|\mathcal{H}| \leq \binom{n+1}{m} - \binom{a}{m-1}, \tag{9}$$

and equality holds if and only if

(i) $\mathcal{H} = [n+1]^{(m)} \setminus \big(A^{(m-1)} \times \{n+1\}\big)$, *or*

(ii) $\mathcal{H} = [n+1]^{(m-1)} \setminus \big(A^{(m-2)} \times \{n+1\}\big)$,
where n is even, $k > n/2$, and $a \leq n/2-2$ or $a = n-1$.

Proof Among all antichains $\mathcal{H}$ as in the lemma, consider one of maximum cardinality. Let $\ell := \min\{|H| : H \in \mathcal{H}\}$ and $u := \max\{|H| : H \in \mathcal{H}\}$, and $\mathcal{H}_i := \{X \in \mathcal{H} : |X| = i\}$ for $\ell \leq i \leq u$.

Assume that $u \geq (n+3)/2$. It is straight forward to verify that $\mathcal{H}' := (\mathcal{H} \setminus \mathcal{H}_u) \cup \tilde\Delta\mathcal{H}_u$ is an antichain that also satisfies the conditions in the theorem. By Lemma 12(b), we have $|\mathcal{H}'| > |\mathcal{H}|$, a contradiction to the maximality of $|\mathcal{H}|$. Hence,

$$u \leq (n+2)/2. \tag{10}$$

Next, assume that $\ell < \min\{n/2, k\}$. For $n = 2$ this means $\ell = 0$ and thus $\mathcal{H} = \{\emptyset\}$ which is clearly not of maximal cardinality. Otherwise consider $\mathcal{H}'' := (\mathcal{H} \setminus \mathcal{H}_\ell) \cup \tilde\nabla\mathcal{H}_\ell$ which satisfies the conditions in the lemma. By Lemma 12(a), we have $|\mathcal{H}''| > |\mathcal{H}|$, a contradiction. Consequently,

$$\ell \geq \min\{n/2, k\}. \tag{11}$$

If $k \leq n/2$ then $\mathcal{H} \subseteq [n+1]^{(m)}$ follows from (11) and $\mathcal{H} \subseteq [n+1]^{(\leq k)}$. If $k > n/2$ and n is odd then $\mathcal{H} \subseteq [n+1]^{(m)}$ by (10) and (11). By the maximality of $|\mathcal{H}|$, it follows that $\mathcal{H}$ must be the antichain given in (i) for which equality in (9) is obviously attained.

That leaves $k > n/2$ and n is even. Here (10) and (11) imply $\mathcal{H} = \mathcal{H}_{n/2} \cup \mathcal{H}_{n/2+1}$. By the maximality of $|\mathcal{H}|$, we have $|\tilde{\nabla}\mathcal{H}_{n/2}| \leq |\mathcal{H}_{n/2}|$. According to Lemma 12(a), this is only possible if $\mathcal{H}_{n/2} = \emptyset$ or if $\mathcal{H}_{n/2} = [n+1]^{(n/2)} \setminus \big(A^{(n/2-1)} \times \{n+1\}\big)$, where $a \leq n/2-2$ or $a = n-1$. In the first case, $\mathcal{H}$ must be as in (i). In the latter case, $\mathcal{H}_{n/2+1}$ must be empty because $\mathcal{H}$ is an antichain, and $\mathcal{H}$ is as in (ii). Finally, it is straightforward to verify that for $\mathcal{H}$ as in (ii) equality is attained in (9).

Proof of Theorem 11.

Let $\mathcal{G}^* := \{G \in \mathcal{G} : G \subset F \text{ for some } F \in \mathcal{F}\}$. Then

$$\mathcal{H} := \mathcal{F} \cup (\mathcal{G} \setminus \mathcal{G}^*) \cup \big\{G \cup \{n+1\} : G \in \mathcal{G}^*\big\}$$

has the same cardinality as $\mathcal{F} \cup \mathcal{G}$ and satisfies the conditions in Lemma 13. Now (9) implies the bound (1).

It is easy to verify that any $\mathcal{F}$ and $\mathcal{G}$ as in (i) or (ii) satisfy the conditions in Theorem 11, and that equality in (1) is attained for such families. It remains to show that these are the only optimal choices of $\mathcal{F}$ and $\mathcal{G}$.

For equality to be attained in (1), we must have equality in (9), i.e., $\mathcal{H}$ must be as in Lemma 13 (i) or (ii).

If $\mathcal{H}$ is as in Lemma 13 (i), then the definition of $\mathcal{H}$ implies

$$\mathcal{F} \cup \mathcal{G} = [n]^{(m)} \cup ([n]^{(m-1)} \setminus A^{(m-1)})$$

and $[n]^{(m-1)} \setminus A^{(m-1)} = \mathcal{G}^* \subseteq \mathcal{G}$. Similarly, if $\mathcal{H}$ is as in Lemma 13 (ii), then $\mathcal{F}$ and $\mathcal{G}$ are as in (ii).

Furthermore, for $k > 1$ we have $m > 1$, so $\mathcal{G}^*$ is non-empty. As $\mathcal{G}$ is an antichain, we get $[n]^{(m)} \setminus A^{(m)} \subseteq \mathcal{F}$ in case (i), and $[n]^{(m-1)} \setminus A^{(m-1)} \subseteq \mathcal{F}$ in case (ii). ∎

A.4 Section 6

Theorem 23 (Theorem 12 restated) *Let t be a tuple on (T, T_S) and $\mathfrak{I}_{c\langle X\rangle}$ a certain-key-index for table I over (T, T_S). Define*

$$K := \{A \in X \mid t[A] \neq \bot\}$$

Then existence of a tuple in I weakly similar to t can be checked with 2^k lookups in $\mathfrak{I}_K$, where $k := |K \setminus T_S|$.

Proof A tuple t' is weakly similar to t on X iff it is weakly similar to t on K. We have

$$\begin{aligned} t[K] \sim_w t'[K] \quad \Leftrightarrow \quad & t[K \cap T_S] = t'[K \cap T_S] \\ & \wedge \forall A \in K \setminus T_S.\ t'[A] \in \{t[A], \bot\} \end{aligned}$$

which means there are precicely 2^k distinct values $t'[K]$ may take to be weakly similar to t. For each fixed value $t'[K]$, the existence of such a $t' \in I$ can be decided with a single lookup in $\mathfrak{I}_K$.

B Weak and Strong FDs do not enjoy Armstrong relations

Note that there is a "strong similarity" between possible and certain keys and the weak and strong functional dependencies discussed in [33]. A weak functional dependency holds on some possible world, while a strong functional dependency holds on every possible

world. Despite this, [33, Theorem 6.1] claims that for a set of strong and weak functional dependencies, an Armstrong relation always exists. Unfortunately that claim is wrong, with the proof not considering all tuple pairs in Case 2, "if" direction.

We must note here that Armstrong tables in [33] correspond to our pre-Armstrong tables, they do not consider `NOT NULL` constraints, and a weak/strong FD $X \to R$ may hold on a relation over R while the corresponding possible/certain key $_{(p/c)}\langle X \rangle$ does not. Hence non-existence of a (pre-)Armstrong table for a set of possible and certain keys does not imply non-existence of an Armstrong table for the corresponding set of weak and strong FDs.

Thus, to show that the claim in [33, Theorem 6.1] is wrong, rather than just its proof, we provide a counter example next. Following the notation of [33], we denote weak FDs by $\Diamond(X \to Y)$ and strong FDs by $\Box(X \to Y)$.

Example 15 (no FD (pre-)Armstrong table)
Let $T = ABCDE$ and

$$\Sigma = \left\{ \begin{array}{l} \Box(AB \to E), \Box(CD \to E), \\ \Diamond(AC \to E), \Diamond(AD \to E), \\ \Diamond(BC \to E), \Diamond(BD \to E), \\ \Box(E \to ABCD) \end{array} \right\}$$

Note the similarity to Example 10. Now any Armstrong table I must disprove the strong FDs

$$\Box(AC \to E), \Box(AD \to E), \Box(BC \to E), \Box(BD \to E)$$

while respecting the weak FDs

$$\Diamond(AC \to E), \Diamond(AD \to E), \Diamond(BC \to E), \Diamond(BD \to E)\ .$$

Note that I may not contain $\perp$ in column E, as otherwise $\Box(E \to ABCD)$ forces all tuples to be strongly similar on $ABCD$, causing various non-implied FDs to hold.

In each of the four cases, we require two tuples t, t' which violate a strong FD $\Box(X \to E)$ but satisfy $\Diamond(X \to E)$. Thus

$$\begin{array}{ccc} t[X] \sim_w t'[X] & \textit{and} & t[E] \not\sim_s t'[E] \\ t[X] \not\sim_s t'[X] & \textit{or} & t[E] \sim_w t'[E] \end{array}$$

Since $t[E], t'[E] \neq \perp$ they cannot be weakly similar without being strongly similar. Hence we have $t[X] \not\sim_s t'[X]$, meaning either $t[X]$ or $t'[X]$ must contain $\perp$.

Using the same arguments as in Example 10, we obtain (e.g.) $t_A[AB] \sim_w t_B[AB]$ with $t_A[AB]$ and/or $t_B[AB]$ containing $\perp$. From $\Box(AB \to E), \Box(E \to ABCD)$ it then follows that $t_A[AB] \sim_s t_B[AB]$, contradicting the presence of the $\perp$ marker.

C Prefix indexing

The indexing approach described in Section 6 works for any type of index (hash, B-tree, ...), as we only require fast lookup of exact matches. Some index structures however, in

particular B-tree and its variants, also allow efficient lookup of prefix matches. That is, an index over $ABCD$ also supports lookups on A, AB and ABC.

If such index structures are available (and they are in most DBMSs), we can exploit this to reduce the number of indexes required. In particular, if $X = YA$ contains only a single `NULL` attribute A, we can support $c\langle X \rangle$ with only a single prefix index on YA, as this allows lookups for both Y and YA. In case $X = YAB$ contains two `NULL` attributes A and B, we can support $c\langle X \rangle$ with two prefix indices on YAB and YB.

While such an approach cannot prevent an exponential growth in the number of indices required to achieve performance guarantees (each prefix index only supports a linear number of prefix sets), it significantly reduces the number of indices required in practice.