Justification for Inclusion Dependency Normal Form

Mark Levene
Department of Computer Science
University College London
Gower Street
London WC1E 6BT, U.K.
email: mlevene@cs.ucl.ac.uk

Millist W. Vincent
Advanced Computing Research Centre
School of Computer and Information Science
University of South Australia
Adelaide, Australia 5095
email: millist.vincent@unisa.edu.au

Abstract

Functional dependencies (FDs) and inclusion dependencies (INDs) are the most fundamental integrity constraints that arise in practice in relational databases. In this paper we address the issue of normalisation in the presence of FDs and INDs and in particular the semantic justification for *Inclusion Dependency Normal Form* (IDNF), a normal form which combines Boyce-Codd normal form with the restriction on the INDs that they be noncircular and key-based. We motivate and formalise three goals of database design in the presence of FDs and INDs: non interaction between FDs and INDs, elimination of redundancy and update anomalies, and preservation of entity integrity. We show that, as for FDs, in the presence of INDs being free of redundancy is equivalent to being free of update anomalies. Then, for each of these properties we derive equivalent syntactic conditions on the database design. Individually, each of these syntactic conditions is weaker than IDNF and the restriction that an FD not be embedded in the right hand-side of an IND is common to three of the conditions. However, we also show that for these three goals of database design to be satisfied simultaneously IDNF is both a necessary and sufficient condition.

Keywords relational database design, normal forms, functional dependency, inclusion dependency

1 Introduction

Functional dependencies (FDs) [Arm74, Mai83, Ull88, AA93] generalise the notions of *entity integrity* and *keys* [Cod79] and inclusion dependencies (INDs) [Mit83, CFP84] generalise the notions of *referential integrity* and *foreign keys* [Cod79, Dat86]. In this sense FDs and INDs are the most fundamental data dependencies that arise in practice.

Relational database design in the presence of FDs is an established area in database theory, which has been researched for over twenty years [Cod74, BB79, Mai83, Ull88, AA93]. The semantic justification of the normal forms in the presence of FDs is well-understood in terms of eliminating the so called update anomalies and redundancy problems that can arise in a relation satisfying a set of FDs [BG80, Fag81, Cha89, Vin94, Vin98]. The advice that is given as a result of this investigation of the semantics of the normal forms is that in order to eliminate the above mentioned problems we should design database schemas which are in *Boyce-Codd Normal Form* (BCNF) [Cod74].

Despite the importance of INDs as integrity constraints little research has been carried out on how they should be integrated into the normalisation process of a relational database. Such an integration is fundamental to the success of a design, since the enforcement of referential integrity is no simple matter [CTF88]. Normal forms which include FDs and INDs have been considered in [CA84, MR86, LG92, MR92, BD93] but necessary and sufficient conditions in terms of removing the update anomalies and redundancy problems were not given. It is our goal in this paper to fill in this gap by providing sufficient and necessary semantics for *Inclusion Dependency Normal Form* (IDNF).

We consider some of the problems that occur in the presence of FDs and INDs through two examples. The first example illustrates the situation when an attribute is redundant due to interaction between FDs and INDs.

Example 1.1 Let HEAD be a relation schema, with attributes H and D, where H stands for head of department and D stands for department, and let LECT be a relation schema, with attributes L and D, where L stands for lecturer and as before D stands for department. Furthermore, let F = $\{$HEAD : H $\rightarrow$ D, LECT : L $\rightarrow$ D$\}$ be a set of FDs over a database schema $\mathbf{R}$ = $\{$HEAD, LECT$\}$, stating that a head of department manages a unique department and a lecturer works in a unique department, and I = $\{$HEAD[HD] $\subseteq$ LECT[LD]$\}$ be a set of INDs over $\mathbf{R}$ stating that a head of department also works as a lecturer in the same department. We note that I $\cup$ (F $-$ $\{$HEAD : H $\rightarrow$ D$\}$) $\models$ HEAD : H $\rightarrow$ D, where $\models$ denotes logical implication, by the *pullback* inference rule (see Proposition 2.1), and thus the FD HEAD : H $\rightarrow$ D in F is redundant. Also, note that we have *not* assumed that HEAD : D $\rightarrow$ H in F and thus a department may have more than one head.

Two problems arise with respect to $\mathbf{R}$ and F $\cup$ I. Firstly, the interaction between F and I may lead to the logical implication of data dependencies that were not envisaged by the database designer and may not be easy to detect; in general the implication problem for FDs and INDs is intractable (see the discussion in Section 2). In this example the pullback inference rule implies that an FD in F is redundant.

Secondly, the IND HEAD[HD] $\subseteq$ LECT[LD] combined with the FD LECT : L $\rightarrow$ D imply that the attribute D in HEAD is redundant, since the department of a head can be inferred from the fact that L is a key for LECT. (Formally this inference can be done with the aid of a relational algebra expression which uses renaming, join and projection; see [Mai83, Ull88, AA93] for details on the relational algebra.) Thus HEAD[HD] $\subseteq$ LECT[LD] can be replaced by HEAD[H] $\subseteq$ LECT[L] and the attribute D in HEAD can be removed without any loss of information.

The second example illustrates the situation when the propagation of insertions due to INDs may result in the violation of entity integrity.

Example 1.2 Let EMP be a relation schema, with attributes E and P, where E stands for employee name and P stands for project title, and let PROJ be a relation schema, with attributes P and L, where as before P stands for project title and L stands for project location. Furthermore, let F = $\{$EMP : E $\rightarrow$ P$\}$ be a set of FDs over a database schema $\mathbf{R}$ = $\{$EMP, PROJ$\}$, stating that an employee works on a unique project, and I = $\{$EMP[P] $\subseteq$ PROJ[P]$\}$ be a set of INDs over $\mathbf{R}$ stating that an employee's project is one of the listed projects. We note that a project may be situated in several locations and correspondingly a location may be associated with several projects and thus $\{$P, L$\}$ is the primary key of PROJ.

The problem that arises with respect to $\mathbf{R}$ and F $\cup$ I is that the right-hand side, P, of the IND EMP[P] $\subseteq$ PROJ[P] is a proper subset of the primary key of PROJ. Let r_1 and r_2 be relations over EMP and PROJ, respectively. Suppose that an employee is assigned to a new project which has not yet been allocated a location and is thus not yet recorded in r_2. Now, due to the IND in I, the insertion of the employee tuple into r_1, having this new project, should be propagated to r_2 by inserting into r_2 a tuple recording the new project. But since the location of the project is still unknown, then due to entity integrity, it is not possible to propagate this insertion to r_2.

We summarise the problems that we would like to avoid when designing relational databases in the presence of FDs and INDs. Firstly, we should avoid redundant attributes, secondly we should avoid the violation of entity integrity when propagating insertions, and lastly we should avoid any interaction between FDs and INDs due to the intractability of the joint implication problem for

FDs and INDs. The main contributions of this paper are the formalisation of these design problems and the result that if the database schema is in IDNF then all of these problems are eliminated. We also demonstrate the robustness of IDNF by showing that, as for BCNF, removing redundancy from the database schema in the presence of FDs and INDs is also equivalent to eliminating update anomalies from the database schema.

The layout of the rest of the paper is as follows. In Section 2 we formally define FDs, INDs and their satisfaction, and introduce the chase procedure as a means of testing and enforcing the satisfaction of a set of FDs and INDs. In Section 3 we formalise the notion of no interaction between a set of FDs and INDs and give a necessary and sufficient condition for FDs and noncircular INDs not to interact. In Section 4 we characterise redundancy in the presence of FDs and INDs. In Section 5 we characterise insertion and modification anomalies in the presence of FDs and INDs and show an equivalence between being free of either insertion or modification anomalies and being free of redundancy. In Section 6 we characterise a generalisation of entity integrity in the presence of FDs and INDs. In Section 7 we define IDNF and present our main result that establishes the semantics of IDNF in terms of either the update anomalies or redundancy problems, and the satisfaction of generalised entity integrity. Finally, in Section 8 we give our concluding remarks and indicate our current research direction.

Definition 1.1 (Notation) We denote the cardinality of a set S by |S|. The *size* of a set S is defined to be the cardinality of a standard encoding of S.

If S is a subset of T we write S $\subseteq$ T and if S is a proper subset of T we write S $\subset$ T. We often denote the singleton {A} simply by A and write A $\in$ S to mean {A} $\subseteq$ S. In addition, we often denote the union of two sets S, T, i.e. S $\cup$ T, simply by ST.

2 Functional and inclusion dependencies

We formalise the notions of FDs and INDs and their satisfaction, and define some useful subclasses of FDs and INDs. We also present the chase procedure for testing and enforcing the satisfaction of a set of FDs and INDs. The chase procedure is instrumental in proving our main results.

Definition 2.1 (Database schema and database) Let U be a finite set of attributes. A *relation schema* R is a finite sequence of distinct attributes from U. A *database schema* is a finite set $\mathbf{R} = \{R_1, R_2, \ldots, R_n\}$, such that each $R_i \in \mathbf{R}$ is a relation schema and $\bigcup_i R_i = U$.

We assume a countably infinite domain of values, $\mathcal{D}$; without loss of generality we assume that $\mathcal{D}$ is linearly ordered. An R-tuple (or simply a tuple whenever R is understood from context) is a member of the Cartesian product $\mathcal{D} \times \ldots \times \mathcal{D}$ (|R| times).

A *relation* r over R is a finite (possibly empty) set of R-tuples. A *database* d over $\mathbf{R}$ is a family of n relations $\{r_1, r_2, \ldots, r_n\}$ such that each $r_i \in d$ is over $R_i \in \mathbf{R}$. Given a tuple t over R and assuming that $r \in d$ is the relation in d over R, we denote the insertion of t into r by $d \cup \{t\}$, and the deletion of t from r by $d - \{t\}$.

From now on we let $\mathbf{R}$ be a database schema and d be a database over $\mathbf{R}$. Furthermore, we let $r \in d$ be a relation over the relation schema R $\in \mathbf{R}$.

Definition 2.2 (Projection) The *projection* of an R-tuple t onto a set of attributes Y $\subseteq$ R, denoted by t[Y] (also called the Y-value of t), is the restriction of t to Y. The projection of a relation r onto Y, denoted as $\pi_{\mathrm{Y}}(r)$, is defined by $\pi_{\mathrm{Y}}(r) = \{t[\mathrm{Y}] \mid t \in r\}$.

Definition 2.3 (Functional Dependency) A *functional dependency* (or simply an FD) over a database schema $\mathbf{R}$ is a statement of the form R : X $\rightarrow$ Y (alternatively X $\rightarrow$ Y is an FD over the relation schema R), where R $\in$ **R** and X, Y $\subseteq$ R are sets of attributes. An FD of the form R : X $\rightarrow$ Y is said to be *trivial* if Y $\subseteq$ X; it is said to be *standard* if X $\neq \emptyset$.

An FD R : X $\rightarrow$ Y is satisfied in d, denoted by $d \models$ R : X $\rightarrow$ Y, whenever $\forall t_1, t_2 \in r$, if t_1[X] = t_2[X] then t_1[Y] = t_2[Y].

Definition 2.4 (Inclusion Dependency) An *inclusion dependency* (or simply an IND) over a database schema **R** is a statement of the form R_i[X] $\subseteq$ R_j[Y], where $R_i, R_j \in$ **R** and X $\subseteq R_i$, Y $\subseteq R_j$ are sequences of distinct attributes such that |X| = |Y|. An IND is said to be *trivial* if it is of the form R[X] $\subseteq$ R[X]. An IND R[X] $\subseteq$ S[Y] is said to be unary if |X| = 1. An IND is said to be *typed* if it is of the form R[X] $\subseteq$ S[X].

An IND R_i[X] $\subseteq$ R_j[Y] over **R** is satisfied in d, denoted by d $\models$ R_i[X] $\subseteq$ R_j[Y], whenever $\pi_{\mathrm{X}}(r_i) \subseteq \pi_{\mathrm{Y}}(r_j)$, where $r_i, r_j \in d$ are the relations over R_i and R_j, respectively.

From now on we let F be a set of FDs over **R** and $F_i = \{R_i : \mathrm{X} \rightarrow \mathrm{Y} \in \mathrm{F}\}$, $\{1, 2, \ldots, n\}$, be the set of FDs in F over $R_i \in$ **R**. Furthermore, we let I be a set of INDs over **R** and let $\Sigma = \mathrm{F} \cup \mathrm{I}$.

Definition 2.5 (Logical implication) Σ is *satisfied* in d, denoted by $d \models \Sigma$, if $\forall \sigma \in \Sigma, d \models \sigma$.

Σ *logically implies* an FD or an IND σ, written $\Sigma \models \sigma$, if whenever d is a database over **R** then the following condition is true:

if $d \models \Sigma$ holds then $d \models \sigma$ also holds.

Σ *logically implies* a set Γ of FDs and INDs over **R**, written $\Sigma \models \Gamma$, if $\forall \sigma \in \Gamma, \Sigma \models \sigma$. We let Σ^+ denote the set of all FDs and INDs that are logically implied by Σ.

Definition 2.6 (Keys, BCNF and key-based INDs) A set of attributes X $\subseteq R_i$ is a *superkey* for R_i with respect to F_i if $F_i \models R_i$: X $\rightarrow R_i$ holds; X is a *key* for R_i with respect to F_i if it is a superkey for R_i with respect to F_i and for no proper subset Y $\subset$ X is Y a superkey for R_i with respect to F_i. We let KEYS(F) be the set of all FDs of the form X $\rightarrow R_i$, where X is a key for R_i with respect to F_i, for $i \in \{1, 2, \ldots, n\}$.

A database schema **R** is in *Boyce-Codd Normal Form* (or simply BCNF) with respect to F if for all $R_i \in$ **R**, for all nontrivial FDs R_i : X $\rightarrow$ Y $\in F_i$, X is a superkey for R_i with respect to F_i.

An IND R_i[X] $\subseteq$ R_j[Y] is *superkey-based*, respectively *key-based*, if Y is a superkey, respectively a key, for R_j with respect to F_j.

Definition 2.7 (Circular and noncircular sets of INDs) A set I of INDs over **R** is *circular* if either

1. I contains a nontrivial IND R[X] $\subseteq$ R[Y], or
2. there exist m distinct relation schemas, $R_1, R_2, R_3, \ldots, R_m \in$ **R**, with $m > 1$, such that I contains the INDs: $R_1[X_1] \subseteq R_2[Y_2]$, $R_2[X_2] \subseteq R_3[Y_3]$, ..., $R_m[X_m] \subseteq R_1[Y_1]$.

A set of INDs I is *noncircular* if it is not circular.

The class of proper circular INDs [Imi91] defined below includes the class of noncircular INDs as a special case.

Definition 2.8 (Proper circular sets of INDs) A set I of INDs over $\mathbf{R}$ is *proper circular* if it is either noncircular or whenever there exist m distinct relation schemas, $R_1, R_2, R_3, \ldots, R_m \in \mathbf{R}$, with $m > 1$, such that I contains the INDs: $R_1[X_1] \subseteq R_2[Y_2]$, $R_2[X_2] \subseteq R_3[Y_3]$, …, $R_{m-1}[X_{m-1}] \subseteq R_m[Y_m]$, $R_m[X_m] \subseteq R_1[Y_1]$, then for all $i \in \{1, 2, \ldots, m\}$ we have $X_i = Y_i$.

It is well known that Armstrong's axiom system [Arm74, Mai83, Ull88, AA93] can be used to compute F^+ and that Casanova's et al. axiom system [CFP84] can be used to compute I^+. However, when we consider FDs and INDs together computing Σ^+ was shown to be undecidable [Mit83, CV85]. On the other hand, when I is noncircular then Mitchell's axiom system [Mit83] can be used to compute Σ^+ [CK86]. Moreover, in the special case when I is a set of unary INDs then Cosmadakis's et al. axiom system [CKV90] can be used to compute Σ^+.

The implication problem is the problem of deciding whether $\sigma \in \Sigma^+$, where σ is an FD or IND and Σ is a set of FDs and INDs. It is well-know that the implication problem for FDs on their own is decidable in linear time [BB79]. On the other hand the implication problem for INDs is, in general, PSPACE-complete [CFP84]. The implication problem for noncircular INDs is NP-complete [Man84, CK86]. Typed INDs have a polynomial time implication problem [CV83]. Unary INDs have a linear time implication problem [CKV90]. When we consider FDs and INDs together the implication problem is undecidable, as mentioned above. The implication problem for FDs and noncircular INDs is EXPTIME-complete and if the noncircular INDs are typed then the implication problem is NP-hard [CK86]. FDs and unary INDs have a polynomial time implication problem [CKV90].

The next proposition describes the *pullback* inference rule [Mit83, CFP84], which allows us to infer an FD from an FD and an IND.

Proposition 2.1 If $\Sigma \models \{\mathrm{R[XY]} \subseteq \mathrm{S[WZ]}, \mathrm{S} : \mathrm{W} \rightarrow \mathrm{Z}\}$ and $|\mathrm{X}|=|\mathrm{W}|$, then $\Sigma \models \mathrm{R} : \mathrm{X} \rightarrow \mathrm{Y}$. □

Definition 2.9 (Reduced set of FDs and INDs) The projection of a set of FDs F_i over R_i onto a set of attributes $\mathrm{Y} \subseteq R_i$, denoted by $F_i[\mathrm{Y}]$, is given by $F_i[\mathrm{Y}] = \{R_i : \mathrm{W} \rightarrow \mathrm{Z} \mid R_i : \mathrm{W} \rightarrow \mathrm{Z} \in F_i^+ \text{ and } \mathrm{WZ} \subseteq \mathrm{Y}\}$.

A set of attributes $\mathrm{Y} \subseteq R_i$ is said to be *reduced* with respect to R_i and a set of FDs F_i over R_i (or simply reduced with respect to F_i if R_i is understood from context) if $F_i[\mathrm{Y}]$ contains only trivial FDs. A set of FDs and INDs $\Sigma = \mathrm{F} \cup \mathrm{I}$ is said to be *reduced* if $\forall\ R_i[\mathrm{X}] \subseteq R_j[\mathrm{Y}] \in \mathrm{I}$, Y is reduced with respect to F_j.

It can easily be shown that it can be decided in polynomial time in the size of Σ whether Σ is reduced or not.

The chase procedure provides us with an algorithm which forces a database to satisfy a set of FDs and INDs.

Definition 2.10 (The chase procedure for INDs) The chase of d with respect to Σ, denoted by CHASE(d, Σ), is the result of applying the following chase rules, that is the FD and the IND rules, to the current state of d as long as possible. (The current state of d prior to the first application of the chase rule is its state upon input to the chase procedure.)

FD rule: If $R_j : \mathrm{X} \rightarrow \mathrm{Y} \in F_j$ and $\exists t_1, t_2 \in r_j$ such that $t_1[\mathrm{X}] = t_2[\mathrm{X}]$ but $t_1[\mathrm{Y}] \neq t_2[\mathrm{Y}]$ then, $\forall\ \mathrm{A} \in \mathrm{Y}$, change all the occurrences in d of the larger of the values of $t_1[\mathrm{A}]$ and $t_2[\mathrm{A}]$ to the smaller of the values of $t_1[\mathrm{A}]$ and $t_2[\mathrm{A}]$.

IND rule: If $R_i[\mathrm{X}] \subseteq R_j[\mathrm{Y}] \in \mathrm{I}$ and $\exists t \in r_i$ such that $t[\mathrm{X}] \notin \pi_{\mathrm{Y}}(r_j)$, then add a tuple u over R_j to r_j, where $u[\mathrm{Y}] = t[\mathrm{X}]$ and $\forall$ A $\in R_j - \mathrm{Y}$, $u[\mathrm{A}]$ is assigned a new value greater than any other current value occurring in the tuples of relations in the current state of d.

We observe that if we allow I to be circular then the chase procedure does not always terminate [JK84]. (When the chase of d with respect to Σ does not terminate, then CHASE(d, Σ) is said to violate a set of FDs G over **R**, i.e. CHASE$(d, \Sigma) \not\models$ G, if after some finite number of applications of the IND rule to the current state of d, resulting in d', we have that $d' \not\models$ G.) In the special case when I is in the class of *proper circular* INDs then it was shown that the chase procedure always terminates [Imi91].

The following theorem is a consequence of results in [MR92, Chapter 10].

Theorem 2.2 Let $\Sigma = \mathrm{F} \cup \mathrm{I}$ be a set of FDs and proper circular INDs over a database schema **R**. Then the following two statements are true:

(i) CHASE$(d, \Sigma) \models \Sigma$.

(ii) CHASE(d, Σ) terminates after a finite number of applications of the IND rule to the current state of d. $\square$

3 Interaction between FDs and INDs

As demonstrated by Proposition 2.1 FDs and INDs may interact in the sense that there may be FDs and INDs implied by a set of FDs and INDs which are not implied by the FDs or INDs taken separately. From the point of view of database design, interaction is undesirable since a database design may be normalised with respect to the set of FDs but not with respect to the combined set of FDs and INDs. This is illustrated in the following example.

Example 3.1 Consider the database schema **R** = {R, S} where R = S = ABC and a set Σ of FDs F and INDs I over **R** given by F = {S : A $\rightarrow$ BC} and I = {R[AB] $\subseteq$ S[AB]}. It can easily be verified that **R** in BCNF with respect to F. However, if we augment F with the FD R : A $\rightarrow$ B, which is logically implied by Σ on using Proposition 2.1, then **R** is not in BCNF with respect to the augmented set of FDs F, since A is not a superkey for R with respect to the set of FDs {R : A $\rightarrow$ B}.

The other major difficulty that occurs in database design when the set of FDs and INDs interact is a result of the fact that, as noted earlier, the implication problem for an arbitrary set of FDs and INDs is undecidable [Mit83, CV85]. Because of this, in general it cannot be determined whether a database design is in BCNF with respect to an arbitrary set of FDs and INDs, since the set of all logically implied FDs cannot be effectively computed. As a consequence, a desirable goal of database design is that the set of FDs and INDs do not interact. We now formalise the notion of non interaction and characterise, for proper circular INDs, a special case when this non interaction occurs.

Definition 3.1 (No interaction occurring between FDs and INDs) A set of FDs F over **R** is said *not to interact* with of set of INDs I over **R**, if

1) for all FDs α over **R**, for all subsets G $\subseteq$ F, G $\cup$ I $\models \alpha$ if and only if G $\models \alpha$, and

2) for all INDs β over **R**, for all subsets J $\subseteq$ I, F $\cup$ J $\models \beta$ if and only if J $\models \beta$.

The following theorem is proven in [LL95] (see [MR92, Chapter 10]).

Theorem 3.1 If $\mathbf{R}$ is in BCNF with respect to a set of FDs F over $\mathbf{R}$ I is a proper circular set of of INDs over $\mathbf{R}$ and $\Sigma = \mathrm{F} \cup \mathrm{I}$ is reduced then F and I do not interact. $\square$

As the next example shows we cannot, in general, extend Theorem 3.1 to the case when the set of INDs I is not proper circular. In particular, by Proposition 2.1 Σ being reduced is a necessary condition for no interaction to occur between F and I, but it is not a sufficient condition for non interaction.

Example 3.2 Consider a database schema $\mathbf{R} = \{\mathrm{R}, \mathrm{S}\}$, where $\mathrm{R} = \mathrm{S} = \mathrm{AB}$, and a set Σ of FDs F and INDs I over $\mathbf{R}$ given by, $\mathrm{F} = \{\mathrm{R} : \mathrm{A} \rightarrow \mathrm{B}, \mathrm{S} : \mathrm{B} \rightarrow \mathrm{A}\}$ and $\mathrm{I} = \{\mathrm{R}[\mathrm{A}] \subseteq \mathrm{S}[\mathrm{A}], \mathrm{S}[\mathrm{B}] \subseteq \mathrm{R}[\mathrm{B}]\}$. It can easily be verified that Σ is reduced and that I is circular. On using the axiom system of [CKV90], it follows that $\Sigma \models \{\mathrm{R} : \mathrm{B} \rightarrow \mathrm{A}, \mathrm{S} : \mathrm{A} \rightarrow \mathrm{B}\}$ and thus F and I interact.

As another example let $\mathrm{F} = \{\mathrm{R} : \mathrm{A} \rightarrow \mathrm{B}\}$ and $\mathrm{I} = \{\mathrm{R}[\mathrm{A}] \subseteq \mathrm{R}[\mathrm{B}]\}$ It can easily be verified that Σ is reduced and I is circular. Again, F and I interact, since on using the axiom system of [CKV90], $\Sigma \models \{\mathrm{R} : \mathrm{B} \rightarrow \mathrm{A}, \mathrm{R}[\mathrm{B}] \subseteq \mathrm{R}[\mathrm{A}]\}$.

4 Attribute redundancy

In this section we investigate the conditions on database design which ensure the elimination of redundancy, a goal which has been long cited as one of the principal motivations for the use of normalisation in database design [Mai83, Ull88, MR92, AA93]. However it has proven to be somewhat difficult to formalise the intuitive notion of redundancy, and it was only relatively recently that this notion was formalised and its relationship to the classical normal forms was established [Vin94, Vin98]. This definition of redundancy, and the associated normal form that guarantees redundancy elimination, is as follows.

Definition 4.1 (Value redundancy) Let d be a database over $\mathbf{R}$ that satisfies F and $t \in r$ be a tuple, where $r \in d$ is a relation over a relation schema $\mathrm{R} \in \mathbf{R}$. The occurrence of a value $t[\mathrm{A}]$, where $\mathrm{A} \in \mathrm{R}$ is an attribute, is *redundant* in d with respect to F if for every replacement of $t[\mathrm{A}]$ by a distinct value $v \in \mathcal{D}$ such that $v \neq t[\mathrm{A}]$, resulting in the database d', we have that $d' \not\models \mathrm{F}$.

A database schema $\mathbf{R}$ is said to be in *Value Redundancy Free Normal Form* (or simply VRFNF) with respect to a set of FDs F over $\mathbf{R}$ if there does not exist a database d over $\mathbf{R}$ and an occurrence of a value $t[\mathrm{A}]$ that is redundant in d with respect to F.

We now illustrate the definition by a simple example.

Example 4.1 Consider the single relation scheme $\mathbf{R} = \{\mathrm{R}\}$, where $\mathrm{R} = \mathrm{ABC}$, and a set F of FDs given by $\mathrm{F} = \{\mathrm{R} : \mathrm{A} \rightarrow \mathrm{B}\}$. Then R is not in VRFNF, since if we consider the relation r over R, shown in Table 1, then the B-value, 2, present in both tuples, is redundant, since $r \models \mathrm{F}$ and replacing the value 2 in either tuple by another value results in F being violated.

A	B	C
1	2	3
1	2	4

Table 1: The relation r

The following result, which was established in [Vin94, Vin98], shows that given a set of FDs F, VRFNF is equivalent to BCNF. For the sake of completeness, we provide a sketch of the proof.

Theorem 4.1 A database schema $\mathbf{R}$ is in BCNF with respect to F if and only if $\mathbf{R}$ is in VRFNF with respect to F.

Proof. The if part follows by showing the contrapositive that if $\mathbf{R}$ is not in BCNF then it is not in VRFNF. This result follows because if $\mathbf{R}$ is not in BCNF then there exists a non-trivial implied FD X $\rightarrow$ A where X is not a superkey and using a well known construction (Theorem 7.1 in [Ull88]) there exists a two tuple relation r in which the tuples are identical on XA. The relation r is not in VRFNF since changing either of the A-values results in X $\rightarrow$ A being violated. The only if part follows from the observation that if an occurrence t[A] is redundant in a relation r, then there must exist X $\rightarrow$ Y $\in$ F, with A $\in$ Y, and another tuple $t_1 \in r$ such that $t[\mathrm{X}] = t_1[\mathrm{X}]$. This implies that X cannot be a superkey and hence that $\mathbf{R}$ is not in BCNF and so establishes the only if part. □

In [Vin98] it was shown that in the presence of multivalued dependencies 4NF (fourth normal form [Fag77]) is also equivalent to VRFNF. Somewhat surprisingly, the syntactic equivalent for VRFNF, in the most general case where join dependencies are present, is a new normal form that is weaker than PJ/NF (project-join normal form [Fag79]) and 5NF (fifth normal form [Mai83]) [Vin98].

The concept of value redundancy is of little use though, in evaluating database designs in the presence of INDs because of the following result. It demonstrates that no design where the constraints involve nontrivial INDs can be in VRFNF with respect to the set of INDs.

Lemma 4.2 Let Σ = F $\cup$ I be a set of FDs and noncircular INDs over a database schema $\mathbf{R}$. Then $\mathbf{R}$ is not in VRFNF with respect to Σ if I is nonempty and contains at least one nontrivial IND.

Proof. Let $R_i[\mathrm{X}] \subseteq R_j[\mathrm{Y}]$ be a nontrivial IND in I. We construct a database d such that the relation r_i in d over R_i has in it a single tuple t_i containing zeros and every other relation r_k in d is empty. Now, let $d' = \mathrm{CHASE}(d, \Sigma)$ and thus by Theorem 2.2 $d' \models \Sigma$. Due to the noncircularity of I, we have in d' that $r_i = r_i'$, where r_i' is the current state of r_i in d'. Let r_j' be the current state of the relation r_j over R_j in d'. Then the Y-values of the tuple in r_j' must contain zeros, since $d' \models \Sigma$. Thus all of the zeros in the single tuple in r_i' are redundant, since changing any of them results in $R_i[\mathrm{X}] \subseteq R_j[\mathrm{Y}]$ being violated. □

Due to the above lemma, we require only that the design be in VRFNF with respect to the FDs but not with respect to the INDs. However, as noted by others [Sci86, LG92, MR92], an even stronger form of redundancy can occur in a database in the presence of INDs. We refer to this as attribute redundancy, which was illustrated in Example 1.1 given in the introduction.

Definition 4.2 (Attribute redundancy) An attribute A in a relation schema R $\in \mathbf{R}$ is *redundant* with respect to Σ if whenever d is a database over $\mathbf{R}$ which satisfies Σ and $r \in d$ is a nonempty relation over R, then for every tuple $t \in r$, if t[A] is replaced by a distinct value $v \in \mathcal{D}$ such that $v \neq t[\mathrm{A}]$, resulting in the database d', then $d' \not\models \Sigma$.

A database schema $\mathbf{R}$ is said to be in *Attribute Redundancy Free Normal Form* (or simply ARFNF) with respect to a set of FDs and INDs Σ over $\mathbf{R}$ if there does not exists an attribute A in a relation schema R $\in \mathbf{R}$ which is redundant with respect to Σ.

The next example shows that ARFNF is too weak when Σ contains only FDs, highlighting the difference between VRFNF and ARFNF.

Example 4.2 Consider the relation r of Example 4.1, shown in Table 1, and let r' be the result of adding the tuple $t =< 5, 2, 4 >$ to r. Then, it can easily be verified that, although $\mathbf{R}$ is not in VRFNF with respect to F, $\mathbf{R}$ is in ARFNF with respect to F, since if we replace any value in t by a distinct value then the resulting database still satisfies F.

Combining Definitions 4.1 and 4.2 we can define redundancy free normal form.

Definition 4.3 (Redundancy free normal form) A database schema **R** is said to be in *Redundancy Free Normal Form* (or simply RFNF) with respect to a set of FDs and INDs Σ over **R** if it is in VRFNF with respect to F and in ARFNF with respect to Σ.

The next theorem shows that when the set of INDs is noncircular then RFNF is equivalent to the set of FDs and INDs being reduced and to the database schema being in BCNF.

Theorem 4.3 Let Σ = F $\cup$ I be a set of FDs and noncircular INDs over a database schema **R**. Then **R** is in RFNF with respect to Σ if and only if Σ is reduced and **R** is in BCNF with respect to F.

Proof. If. By Theorem 4.1 **R** is in VRFNF with respect to F. So it remains to show that **R** is in ARFNF.

In the proof we utilise a directed graph representation, G_I = (N, E), of the set of INDs I, which is constructed as follows (see [Sci86]). Each relation schema R in **R** has a separate node in N labelled R; we do not distinguish between nodes and their labels. There is an arc (R, S) $\in$ E if and only if there is a nontrivial IND R[X] $\subseteq$ S[Y] $\in$ I. It can easily be verified that there is a path in G_I from R to S if and only if for some IND R[X] $\subseteq$ S[Y] we have I $\models$ R[X] $\subseteq$ S[Y]. Moreover, since I is noncircular, we have that G_I is acyclic.

Let A $\in$ R_i be an attribute, where $R_i \in$ **R** is a relation schema. We construct a database d having a nonempty relation $r_i \in d$, which exhibits the fact that A is nonredundant with respect to Σ.

We first initialise the database d to be a database d_0 as follows. Let $r_i = r_i^0$ have a single tuple t such that for all B $\in R_i-$ A t[B] = 0 and t[A] = 1. All other relations r_k^0 over relation schemas R_k are initialised to be empty in d_0. Therefore, by Theorem 2.2, we have that d_1 = CHASE(d_0, Σ) $\models \Sigma$. Let r_i^1 in d_1 be the current state of r_i. Then, by Theorem 3.1 we have $r_i^1 = r_i$, since $d_0 \models$ F and the current state r_k^1 in d_1 of a relation r_k^0 is empty, if there does not exist a path in G_I from R_i to R_k.

Let $d_2 = (d_1 - \{r_i^1\}) \cup \{r_i^2\}$, where r_i^2 has a single tuple t' such that for all B $\in R_i$, including A, t[B] = 0. Therefore, by Theorem 2.2, we have that d_3 = CHASE(d_2, Σ) $\models \Sigma$. Moreover, as above, $r_i^3 = r_i^2$ where r_i^3 in d_3 is the current state of r_i, since $d_2 \models$ F. Now, let $d = (d_3 - \{r_i^3\}) \cup \{r_i\}$ be the final state of the initialisation of d. Then, $d \models$ F, since $d_3 \models$ F. We claim that it is also that case that $d \models$ I.

Let us call a nontrivial IND R_i[X] $\subseteq R_j$[Y] $\in$ I a *source* IND, if A $\in$ X. By the projection and permutation inference rule for INDs [CFP84] we assume without loss of generality that a source IND has the form R_i[VA] $\subseteq R_j$[WB].

Due to the noncircularity of I any current state r_k in d of a relation r_k^3 is empty, if there does not exist a path in G_I from R_i to R_k; therefore for such r_k we have $r_k = r_k^3 = r_k^2 = r_k^1 = r_k^0 = \emptyset$. Now, if there is an arc from R_i to R_j in G_I, then there is some IND R_i[X] $\subseteq R_j$[Y] in I. There are two cases to consider.

First, if A $\notin$ X then t[X] = t'[X] contains only zeros and thus $d \models R_i$[X] $\subseteq R_j$[Y]. Secondly, if A $\in$ X then R_i[X] $\subseteq R_j$[Y] is a source IND R_i[VA] $\subseteq R_j$[WB]. Let r_j in d be the current state of r_j^3, i.e. $r_j = r_j^3$. Then, t[VA] $\in \pi_{\mathrm{VA}}(r_j)$, since $r_j^0 \subseteq r_j^1 \subseteq r_j^2 \subseteq r_j^3 = r_j$ and t[VA] $\in \pi_{\mathrm{VA}}(r_j^1)$. Therefore $d \models R_i$[X] $\subseteq R_j$[Y]. It follows that for any IND R_i[X] $\subseteq R_j$[Y] such that there is a path from R_i to R_j and such that I $\models R_i$[X] $\subseteq R_j$[Y] we have $d \models R_i$[X] $\subseteq R_j$[Y], since $d_3 \models R_i$[X] $\subseteq R_j$[Y]. Thus $d \models$ I as required. The if part is now concluded since d_3 and d differ only by the replacement of t'[A] = 0 by t[A] = 1.

Only if. By Theorem 4.1 if **R** is not in BCNF with respect to F then it is not in VRFNF. So assuming that Σ is not reduced it remains to show that **R** is not in ARFNF. By this assumption there exists an IND $R_i[\mathrm{X}] \subseteq R_j[\mathrm{Y}] \in \mathrm{I}$ such that $\mathrm{W} \rightarrow \mathrm{B} \in F_j[\mathrm{Y}]$ is a nontrivial FD. By the pullback inference rule there is a nontrivial FD $\mathrm{V} \rightarrow \mathrm{A} \in F_i[\mathrm{X}]$, since $R_i[\mathrm{VA}] \subseteq R_j[\mathrm{WB}]$ by the projection and permutation inference rule for INDs [CFP84]. Now let $t \in r_i$ be a tuple, where $r_i \in d$ is a nonempty relation over R_i, and assume that $d \models \Sigma$. It follows that A is redundant with respect to Σ, since whenever we replace $t[\mathrm{A}]$ by a distinct value resulting in a database d', it can be seen that $d' \not\models R_i[\mathrm{VA}] \subseteq R_j[\mathrm{WB}]$ otherwise $d \not\models R_j : \mathrm{W} \rightarrow \mathrm{B}$ contrary to assumption. □

We next construct two examples which demonstrate that if the conditions of Theorem 4.3 are violated then **R** is not in ARFNF.

Example 4.3 Let **R** be the database schema from Example 1.1 and consider the database d over**R**, shown in Tables 2 and 3, respectively. It can be verified that the set of FDs and INDs for this example is not reduced but that **R** is in BCNF with respect to the set of FDs. The attribute D of the relation schema HEAD can be seen to be redundant, since changing e_1 in Table 2 causes I to be violated. A similar situation occurs for every other database defined over **R**, and so **R** is not in ARFNF.

H	D
h_1	e_1

Table 2: The relation in d over HEAD

L	D
h_1	e_1
l_1	e_1

Table 3: The relation in d over LECT

Example 4.4 Let $\mathbf{R}$ = {STUDENT, ENROL}, be a database schema, with STUDENT = {Stud_ID, Name} and ENROL = {Stud_ID, Course, Address}. Furthermore, let F = {ENROL: Stud_ID→ Address} be a set of FDs over **R** and I = {ENROL[Stud_ID] ⊆ STUDENT[Stud_ID]} be a set of INDs over **R**. It can be verified that the set of FDs and INDs for this example is reduced but that **R** is not in BCNF with respect to the set of FDs. Then **R** is not in RFNF because it is not in VRFNF, since both occurrences of a_1 in ENROL are redundant in the database d shown in Tables 4 and 5, respectively.

Stud_ID	Name
s_1	n_1

Table 4: The relation in d over over STUDENT

Stud_ID	Course	Address
s_1	c_1	a_1
s_1	c_2	a_1

Table 5: The relation in d over ENROL

As the next example shows we cannot extend Theorem 4.3 to the case when the set of INDs I is circular.

Example 4.5 Consider a database schema **R** = {R, S}, where R = AB and S = A, and a set Σ of FDs F and INDs I over **R** given by, F = {R : A → B, R : B → A} and I = {R[A] ⊆ S[A], S[A] ⊆ R[A]}. It can be verified that Σ is reduced, **R** is in BCNF with respect to F and that I is proper circular, unary, typed and also key-based.

Let d be a database over **R** such that $d \models \Sigma$, let $r \in d$ be a nonempty relation over R and let $t \in r$ be a tuple. Assume without loss of generality that d' is the database resulting from replacing $t[\mathrm{A}] = 0$ by a distinct value 1 resulting in a tuple t', with $t'[\mathrm{A}] = 1$. In order to conclude the example we show that $d' \not\models \Sigma$. Assume to the contrary that $d' \models \Sigma$. Thus there must be a tuple

$u \in r$ which is distinct from t and such that $u[\mathrm{A}] = 1$. If this is not the case then $d' \not\models \mathrm{R}[\mathrm{A}] \subseteq \mathrm{S}[\mathrm{A}]$, since $\pi_{\mathrm{A}}(r) = \pi_{\mathrm{A}}(s)$, where $s \in d$ is a relation over S. Moreover, $u[\mathrm{B}] = t[\mathrm{B}] = t'[\mathrm{B}]$, otherwise $d' \not\models \mathrm{R} : \mathrm{A} \rightarrow \mathrm{B}$. It follows that $u[\mathrm{B}] = t[\mathrm{B}]$ but $u[\mathrm{A}] \neq t[\mathrm{A}]$ and thus $d \not\models \mathrm{R} : \mathrm{B} \rightarrow \mathrm{A}$ contrary to assumption. Therefore the attribute $\mathrm{A} \in \mathrm{R}$ must be redundant with respect to Σ. The reader can easily verify that the attribute $\mathrm{A} \in \mathrm{S}$ is also redundant with respect to Σ even if F was empty. It appears that in this example the relation schema S can be removed from **R** without any loss of semantics.

5 Insertion and modification anomalies

In this section, we investigate the conditions under which database design ensures the elimination of key-based update anomalies, (as distinct from other types of update anomalies as investigated in [BG80, Vin94]). This concept was originally introduced in [Fag81] to deal with the insertion and deletion of tuples and was later extended in [Vin92, Vin94] to include the modifications of tuples. A key-based update anomaly is defined to occur when an update to a relation, which can either be an insertion or a deletion or a modification, results in the new relation satisfying key uniqueness but violating some other constraint on the relation. The reason for this being considered undesirable is that the enforcement of key uniqueness can be implemented via relational database software in a much more efficient manner than the enforcement of more general constraints such as FDs [Mai83, Ull88, AA93]. So, if the satisfaction of all the constraints on a relation is a result of key uniqueness then the integrity of the relation after an update can be easily enforced, whereas the existence of a key-based update anomaly implies the converse. Herein we formalise these concepts based on this approach, with the only difference being that because of the presence of INDs we allow an update to propagate to other relations by the chase procedure. We show that being free of insertion anomalies is equivalent to being free of modification anomalies. In addition, we show that when the INDs are noncircular then being free of either insertion or modification anomalies is equivalent to the set of FDs and INDs being reduced and the database schema being in BCNF with respect to the set of FDs. We do not consider deletion anomalies, since in the presence of FDs removing a tuple from a relation that satisfies a set of FDs, does not cause any violation of an FD in the set.

Definition 5.1 (Compatible tuple) A tuple t over R is *compatible* with d with respect to a set of FDs and INDs $\Sigma = \mathrm{F} \cup \mathrm{I}$ over **R** (or simply compatible with d whenever Σ is understood from context), if $d \cup \{t\} \models \mathrm{KEYS}(\mathrm{F})$.

Definition 5.2 (Free of insertion anomalies) A database d over **R** has an *insertion violation* with respect to a set of FDs and INDs $\Sigma = \mathrm{F} \cup \mathrm{I}$ over **R** (or simply d has an insertion violation whenever Σ is understood from context) if

1. $d \models \Sigma$, and
2. there exists a tuple t over R which is compatible with d but $\mathrm{CHASE}(d \cup \{t\}, \mathrm{I}) \not\models \Sigma$.

A database schema **R** is *free of insertion anomalies* with respect to Σ (or simply **R** is free of insertion anomalies if Σ is understood from context) if there does not exist a database d over **R** which has an insertion violation.

We note that in Definition 5.2 we have utilised the chase procedure to enforce the propagation of insertions of tuples due to the INDs in I. As an example of an insertion violation consider the database schema **R** in Example 1.1 and let d be the database where r_1, the relation over HEAD, is empty and r_2, the relation over LECT, contains the single tuple $< 0, 0 >$. Then d has an insertion violation when the tuple $< 0, 1 >$ is inserted into r_1, since applying the chase procedure results in $< 0, 1 >$ being added to the relation r_2 and thus violating the FD L $\rightarrow$ D.

The next theorem shows that assuming that the set of INDs is noncircular, being in BCNF and the set of FDs and INDs being reduced is equivalent to being free of insertion anomalies.

Theorem 5.1 Let Σ = F $\cup$ I be a set of FDs and noncircular INDs over a database schema **R**. Then **R** is free of insertion anomalies if and only if Σ is reduced and **R** is in BCNF with respect to F.

Proof. If. Let d be a database over **R** such that $d \models \Sigma$ and let t over R_i, where $r_i \in$ **R**, be a tuple which is compatible with d. It remains to show that CHASE($d \cup \{t\}$, I) $\models \Sigma$. We first claim that CHASE($d \cup \{t\}$, I) = CHASE($d \cup \{t\}, \Sigma$). This holds due to Theorem 3.1 implying that the FD rule need never be invoked during the computation of CHASE($d \cup \{t\}, \Sigma$). Moreover, $d \cup \{t\} \models$ F, due to the fact that **R** is in BCNF with respect to F and t is compatible with d. So, by Theorem 2.2 we have CHASE($d \cup \{t\}, \Sigma$) $\models \Sigma$ and thus CHASE($d \cup \{t\}$, I) $\models \Sigma$ as required.

Only if. There are two cases to consider.

Case 1. If **R** is not in BCNF with respect to F, then some $R_i \in$ **R** is not in BCNF with respect to F_i. Thus there is a nontrivial FD X $\rightarrow$ Y $\in F_i$ such that X is not a superkey for R_i with respect to F_i. Assume that X is reduced with respect to R_i and F_i, otherwise replace X by a reduced subset W of X such that W $\rightarrow$ X $\in F_i^+$. It follows that X is a proper subset of a superkey of R_i with respect to F_i. Let r_i over R_i contain a single tuple containing zeros and let all other relations in d be empty. We can assume without loss of generality that $d \models \Sigma$, otherwise we let d be CHASE(d, Σ). Due to I being noncircular the state of r_i remains unchanged in CHASE(d, Σ). Now, let t be a tuple whose X-values are zeros and such that all its other values are ones. Then t is compatible with d but CHASE($d \cup \{t\}$, I) $\not\models \Sigma$, since the FD X $\rightarrow$ Y will be violated in the current state of r_i. Therefore, **R** is not free of insertion anomalies.

Case 2. If Σ is not reduced but **R** is in BCNF with respect to F then we have an IND R_i[X] $\subseteq R_j$[Y], where Y is a proper superset of a key, say W, for R_j with respect to F_j. We let r_j over R_j contain a single tuple, say t_j containing zeros and let all other relations in d, including r_i over R_i, be empty. We can assume without loss of generality that $d \models \Sigma$, otherwise we let d be CHASE(d, Σ). Due to I being noncircular the state of r_i remains unchanged in CHASE(d, Σ). Now, let t over R_i be a tuple which agrees with t_j on its W-value but disagrees with t_j on the rest of its values. Then, t is compatible with d but CHASE($d \cup \{t\}$, I) $\not\models \Sigma$, since the FD W $\rightarrow R_j$ will be violated in the resulting current state of r_j. $\square$

To illustrate this theorem we note firstly that the example given before Theorem 5.1, demonstrates the case where a database schema has an insertion anomaly when the set of dependencies is not reduced. Alternatively, the following example demonstrates the case of a database schema not being in BCNF and having an insertion anomaly.

Example 5.1 Let **R**, Σ = F $\cup$ I, be as in Example 4.4. We start with the database d shown in Tables 6 and 7, respectively. If we then insert the tuple $< s_1, c_2, a_2 >$, which is compatible with ENROL, into the ENROL relation, applying the chase procedure results in the database d' shown in Tables 8 and 9, respectively, where n_2 is a new value. It can be seen that d' violates Σ and so **R** is not free of insertion anomalies.

In the next example we show that the only if part of Theorem 5.1 is, in general, false, even when I is a proper circular set of INDs.

Stud_ID	Name
s_1	n_1

Table 6: The relation in d over over STUDENT

Stud_ID	Course	Address
s_1	c_1	a_1

Table 7: The relation in d over ENROL

Stud_ID	Name
s_1	n_1
s_1	n_2

Table 8: The relation in d' over over STUDENT

Stud_ID	Course	Address
s_1	c_1	a_1
s_1	c_2	a_2

Table 9: The relation in d' over ENROL

Example 5.2 Consider a database schema $\mathbf{R} = \{R, S\}$, where R = S = AB, and a set Σ of FDs F and INDs I over $\mathbf{R}$ given by, F = $\{R : A \rightarrow B, S : A \rightarrow B\}$ and I = $\{R[AB] \subseteq S[AB], S[AB] \subseteq R[AB]\}$. It can easily be verified that I is proper circular, $\mathbf{R}$ is in BCNF but that Σ is not reduced. In addition, $\mathbf{R}$ is free of insertion anomalies, since for any database $d = \{r, s\}$ such that $d \models \Sigma$, where r and s are the relations in d over R and S, respectively, we have that $r = s$ due to I. If we drop $S[AB] \subseteq R[AB]$ from I, then, as in the proof of the only if part of Theorem 5.1, $\mathbf{R}$ has an insertion violation.

The next example illustrates that we cannot, in general, extend Theorem 5.1 to the case when the set of INDs I is circular even when Σ is reduced, due to possible interaction between the FDs and INDs.

Example 5.3 Consider a database schema $\mathbf{R} = \{R\}$, where R = AB, and a set Σ of FDs F and INDs I over $\mathbf{R}$ given by, F = $\{R : A \rightarrow B\}$ and I = $\{R[A] \subseteq R[B]\}$. It can be verified that Σ is reduced but I is circular. As was shown in Example 3.2 although Σ is reduced $\Sigma \models \{R : B \rightarrow A, R[B] \subseteq R[A]\}$ and thus F and I interact.

Let d be a database over $\mathbf{R}$ such that the relation r over R contains the single tuple $< 0, 0 >$ and let t be the tuple $< 1, 0 >$. Then, d has an insertion violation, since $d \models \Sigma$, t is compatible with d but CHASE($d \cup \{t\}$, I) $\not\models$ R : B $\rightarrow$ A. (In fact, in this case the chase procedure does not terminate, but since t is inserted into r and the chase procedure does not modify any of the tuples in its input database, then it does not satisfy Σ; see the comment after Definition 2.10.)

We now formally define the second type of key-based update anomaly, a modification anomaly, following the approach in [Vin92, Vin94] with the only difference again being that the chase procedure is used to propagate the effects of the change into other relations.

Definition 5.3 (Free of modification anomalies) A database d over $\mathbf{R}$ has a *modification violation* with respect to a set of FDs and INDs Σ = F $\cup$ I over $\mathbf{R}$ (or simply d has a modification violation whenever Σ is understood from context) if

1. $d \models \Sigma$, and
2. there exists a tuple $u \in r$, where $r \in d$ is the relation over R, and a tuple t over R which is compatible with $d - \{u\}$ but CHASE($(d - \{u\}) \cup \{t\}$, I) $\not\models \Sigma$.

A database schema $\mathbf{R}$ is *free of modification anomalies* with respect to Σ (or simply $\mathbf{R}$ is free of modification anomalies if Σ is understood from context) if there does not exist a database d over $\mathbf{R}$ which has a modification violation.

Theorem 5.2 Let $\Sigma = \mathrm{F} \cup \mathrm{I}$ be a set of FDs and noncircular INDs over a database schema **R**. Then **R** is free of modification anomalies if and only if **R** is free of insertion anomalies.

Proof. If. Let d be a database over **R** such that $d \models \Sigma$, t be a tuple that is compatible with d and $u \in r$ be a tuple, where $r \in d$ is the relation over R. It follows that t is compatible with $d - \{u\}$. We need to show that if CHASE($d \cup \{t\}$, I) $\models \Sigma$ then CHASE($(d - \{u\}) \cup \{t\}$, I) $\models \Sigma$. By Theorem 2.2 of the chase procedure CHASE($(d - \{u\}) \cup \{t\}$, I) $\models$ I, so it remains to show that CHASE($(d - \{u\}) \cup \{t\}$, I) $\models$ F. Let CHASE($d \cup \{t\}$, I) $= \{r_1, r_2, \ldots, r_n\}$ and let CHASE($(d - \{u\}) \cup \{t\}$, I) $= \{s_1, s_2, \ldots, s_n\}$. Then by Definition 2.10 of the chase procedure we have that for all $i \in \{1, 2, \ldots, n\}$, $s_i \subseteq r_i$. It follows that CHASE($(d - \{u\}) \cup \{t\}$, I) $\models \Sigma$, since by Definition 5.2 CHASE($d \cup \{t\}$, I) $\models$ F.

Only if. If **R** is free of modification anomalies then by a similar argument to that made in the *only if* part of the proof of Theorem 5.1 it follows that Σ is reduced and **R** is in BCNF. In the first case we add an additional tuple u over R_i, which contains ones, to the original state of r_i, and in the second case we add an additional tuple u over R_i, which contains zeros, to the original state of r_i. The result now follows by the *if* part of Theorem 5.1. □

Combining Theorems 4.3, 5.1 and 5.2 we obtain the next result.

Corollary 5.3 Let $\Sigma = \mathrm{F} \cup \mathrm{I}$ be a set of FDs and noncircular INDs over a database schema **R**. Then the following statements are equivalent:

(i) **R** is free of insertion anomalies.

(ii) **R** is free of modification anomalies.

(iii) **R** is in RFNF. □

6 Generalised entity integrity

In this section we justify superkey-based INDs on the basis that they do not cause the propagation of the insertion of tuples that represent undefined entities, thus causing the violation of entity integrity. This problem was illustrated in Example 1.2 given in the introduction.

In the next definition we view the chase procedure as a mechanism which enforces the propagation of insertions of tuples due to the INDs in I.

Definition 6.1 (Generalised entity integrity) Let t be a tuple that is added to a relation r_i over R_i, in the current state of a database d, during the computation of CHASE(d, Σ). Then, t is *entity-based* if there exists at least one key X for R_i with respect to F_i such that for all A $\in$ X, t[A] is *not* a new value that is assigned to t as a result of invoking the IND rule.

A database schema **R** satisfies *generalised entity integrity* with respect to a set $\Sigma = \mathrm{F} \cup \mathrm{I}$ of FDs and INDs over **R** if for all databases d over **R**, all the tuples that are added to relations in the current state of d during the computation of CHASE(d, Σ) are entity-based.

The next theorem shows that satisfaction of generalised entity integrity is equivalent to the set of INDs being superkey-based.

Theorem 6.1 A database schema **R** satisfies generalised entity integrity with respect to a set of FDs and INDs $\Sigma = \mathrm{F} \cup \mathrm{I}$ if and only if I is superkey-based.

Proof. If I is superkey-based then the result immediately follows by the definition of the IND rule (see Definition 2.10). On the other hand, if I is not superkey-based then there is some IND R_i[X] $\subseteq R_j$[Y] $\in$ I such that Y is not a superkey for R_j with respect to F_j. Let d be a database over **R** such that all its relations apart for r_i over R_i are empty. The relation r_i has a single tuple. By the definition of the FD rule (see Definition 2.10), we have that for every key, say K, of R_j with respect to F_j there is at least one attribute, say A $\in$ K, such that the tuple t_j added to r_j over R_j is assigned a new A-value by the IND rule, otherwise, contrary to assumption, we can deduce that Y is a superkey for R_j with respect to F_j. It follows that **R** does not satisfy generalised entity integrity, concluding the proof. □

7 Inclusion dependency normal form

A database schema is in IDNF with respect to a set of FDs and INDs if it is in BCNF with respect to the set of FDs and the set of INDs is noncircular and key-based. We show that a database schema is in IDNF if and only if it satisfies generalised entity integrity and is either free of insertion anomalies or free of modification anomalies or in redundancy free normal form.

We next formally define IDNF (cf. [MR86, MR92]).

Definition 7.1 (Inclusion dependency normal form) A database schema **R** is in *Inclusion Dependency Normal Form* (IDNF) with respect to a set of Σ of FDs F and INDs I over **R** (or simply in IDNF if Σ is understood from context) if

1. **R** is in BCNF with respect to F, and
2. I is a noncircular and key-based set of INDs.

We note that if the set of INDs I is empty then **R** being in IDNF is equivalent to **R** being in BCNF. We further note that we have not restricted the FDs in F to be standard.

The next result follows from Corollary 5.3, Theorem 6.1 and Definition 7.1.

Theorem 7.1 Let Σ = F $\cup$ I be a set of FDs and noncircular INDs over a database schema **R**. Then the following statements are equivalent:

(i) **R** is in IDNF

(ii) **R** is free of insertion anomalies and satisfies generalised entity integrity.

(iii) **R** is free of modification anomalies and satisfies generalised entity integrity.

(iv) **R** is in RFNF and satisfies generalised entity integrity. □

8 Concluding Remarks

We have identified three problems that may arise when designing databases in the presence of FDs and INDs, apart from the update anomalies and redundancy problems that may arise in each relation due to the FDs considered on their own. The first problem is that of attribute redundancy, the second problem is the potential violation of entity integrity when propagating insertions, and the third problem concerns avoiding the complex interaction which may occur between FDs and INDs and the intractability of determining such interaction. The first problem was formalised through RFNF and it was shown in Corollary 5.3 that a database schema is in RFNF with respect to a set of FDs and INDs if and only if it is free of insertion anomalies or equivalently free of modification anomalies. This result can be viewed as an extension of a similar result when considering FDs on their own. The second problem was formalised through generalised entity integrity and it was shown in Theorem 6.1 that a database schema satisfies generalised entity integrity with respect to a set of FDs and INDs if and only the set of INDs is superkey-based. The third problem was formalised through the non interaction of the implication problem for FDs and INDs and it was shown in Theorem 3.1 that a set of FDs and INDs do not interact when the set of INDs is proper circular, the set of FDs and INDs are reduced and the database schema is in BCNF. Combining all these result together we obtained, in Theorem 7.1, three equivalent semantic characterisations of IDNF. Theorem 7.1 justifies IDNF as a robust normal form that eliminates both redundancy and update anomalies from the database schema.

If the goal of normalisation is to reduce redundancy then it seems that apart from $\mathbf{R}$ being in BCNF, in general, we must restrict the set of INDs to be noncircular (see Example 4.5). Nonetheless circular sets of INDs arise in practice, for example when we want to express pairwise consistency. (Two relation schemas R and S are *consistent* if the set of INDs I includes the two INDs: $R[R \cap S] \subseteq S[R \cap S]$ and $S[R \cap S] \subseteq R[R \cap S]$. A database schema $\mathbf{R}$ is *pairwise consistent* if every pair of its relation schemas are consistent [BFMY83]; we note that pairwise consistency can be expressed by a set of proper circular INDs.) In this case we need alternative semantics to express the goal of normalisation. A minimal requirement is that the FDs and INDs have no interaction. By Theorem 3.1 as long as the set of FDs and INDs $\Sigma = F \cup I$ is reduced and $\mathbf{R}$ is in BCNF with respect to F then, when the set of INDs I expresses pairwise consistency, F does not interact with I, since I is proper circular. Consider a BCNF database schema $\mathbf{R}$ with relation schemas EMP = {ENAME, DNAME} and DEPT = {DNAME, LOCATION, MGR}, with F = {ENAME $\rightarrow$ DNAME, DNAME $\rightarrow$ {LOCATION, MGR}, MGR $\rightarrow$ DNAME} and I = {EMP[DNAME] $\subseteq$ DEPT[DNAME], DEPT[DNAME] $\subseteq$ EMP[DNAME]}. The set of INDs I expresses the fact that all employees work in departments that exist and all departments have at least one employee. The first IND is key-based but the second is not. Despite this fact it can be verified that F and I do not interact. Moreover, if managers are also employees then we could add the IND DEPT[MGR] $\subseteq$ EMP[ENAME] to I, and it can be verified by exhibiting the appropriate counterexamples that it is still true that F and I do not interact although, now I is not even proper circular. The database schema $\mathbf{R}$ seems to be a reasonable design but it is not in IDNF. Further research needs to be carried out to determine the semantics of normal forms for such FDs and INDs. We conclude the paper by proposing such a normal form. A database schema $\mathbf{R}$ is in *Interaction Free Inclusion Dependency Normal Form* with respect to a set of Σ of FDs F and INDs I over $\mathbf{R}$ if

1. $\mathbf{R}$ is in BCNF with respect to F,
2. All the INDs in I are either key-based or express pairwise consistency, and
3. F and I do not interact.

References

[AA93] P. Atzeni and V. De Antonellis. *Relational Database Theory.* Benjamin/Cummings, Redwood City, Ca., 1993.

[Arm74] W.W. Armstrong. Dependency structures of data base relationships. In *Proceedings of the IFIP Congress*, pages 580–583, Stockholm, 1974.

[BB79] C. Beeri and P.A. Bernstein. Computational problems related to the design of normal form relational schemas. *ACM Transactions on Database Systems*, 4:30–59, 1979.

[BD93] J. Biskup and P. Dublish. Objects in relational database schemes with functional, inclusion and exclusion dependencies. *Theoretical Informatics and Applications*, 27:183–219, 1993.

[BFMY83] C. Beeri, R. Fagin, D. Maier, and M. Yannakakis. On the desirability of acyclic database schemes. *Journal of the ACM*, 30:479–513, 1983.

[BG80] P.A. Bernstein and N. Goodman. What does Boyce-Codd normal form do? In *Proceedings of the International Conference on Very Large Data Bases*, pages 245–259, Montreal, 1980.

[CA84] M.A. Casanova and J.E. Amaral de Sa. Mapping uninterpreted schemes into entity-relationship diagrams: Two applications to conceptual schema design. *IBM Journal of Research and Development*, 28:82–94, 1984.

[CFP84] M.A. Casanova, R. Fagin, and C.H. Papadimitriou. Inclusion dependencies and their interaction with functional dependencies. *Journal of Computer and System Sciences*, 28:29–59, 1984.

[Cha89] E.P.F. Chan. A design theory for solving the anomalies problem. *SIAM Journal on Computing*, 18:429–448, 1989.

[CK86] S.S. Cosmadakis and P.C. Kanellakis. Functional and inclusion dependencies: A graph theoretic approach. In P.C. Kanellakis and F. Preparata, editors, *Advances in Computing Research*, volume 3, pages 163–184. JAI Press, Greenwich, 1986.

[CKV90] S.S. Cosmadakis, P.C. Kanellakis, and M.Y. Vardi. Polynomial-time implication problems for unary inclusion dependencies. *Journal of the ACM*, 37:15–46, 1990.

[Cod74] E.F. Codd. Recent investigations in relational data base systems. In *Proceedings of the IFIP Congress*, pages 1017–1021, Stockholm, 1974.

[Cod79] E.F. Codd. Extending the database relational model to capture more meaning. *ACM Transactions on Database Systems*, 4:397–434, 1979.

[CTF88] M.A. Casanova, L. Tucherman, and A.L. Furtado. Enforcing inclusion dependencies and referential integrity. In *Proceedings of the International Conference on Very Large Data Bases*, pages 38–49, Los Angeles, Ca., 1988.

[CV83] M.A. Casanova and V.M.P. Vidal. Towards a sound view integration methodology. In *Proceedings of ACM Symposium on Principles of Database Systems*, pages 36–47, Atlanta, 1983.

[CV85] A.K. Chandra and M.Y. Vardi. The implication problem for functional and inclusion dependencies is undecidable. *SIAM Journal on Computing*, 14:671–677, 1985.

[Dat86] C.J. Date. Referential integrity. In *Relational Database: Selected Writings*, pages 41–63. Addison-Wesley, Reading, MA, 1986.

[Fag77] R. Fagin. Multivalued dependencies and a new normal form for relational databases. *ACM Transactions on Database Systems*, 2:262–278, 1977.

[Fag79] R. Fagin. Normal forms and relational database operators. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 153–160, Boston, MA, 1979.

[Fag81] R. Fagin. A normal form for relational databases that is based on domains and keys. *ACM Transactions on Database Systems*, 6:387–415, 1981.

[Imi91] T. Imielinski. Abstraction in query processing. *Journal of the ACM*, 38:534–558, 1991.

[JK84] D.S. Johnson and A. Klug. Testing containment of conjunctive queries under functional and inclusion dependencies. *Journal of Computer and System Sciences*, 28:167–189, 1984.

[LG92] T.-W. Ling and C.H. Goh. Logical database design with inclusion dependencies. In *Proceedings of the International Conference on Data Engineering*, pages 642–649, Tempe, Arizona, 1992.

[LL95] M. Levene and G. Loizou. How to prevent interaction of functional and inclusion dependencies. Research Note RN/95/80, Department of Computer Science, University College London, 1995.

[Mai83] D. Maier. *The Theory of Relational Databases.* Computer Science Press, Rockville, MD, 1983.

[Man84] H. Mannila. On the complexity of the inference problem for subclasses of inclusion dependencies. In *Proceedings of the Winter School on Theoretical Computer Science*, pages 182–193, Lammi, Finland, 1984.

[Mit83] J.C. Mitchell. The implication problem for functional and inclusion dependencies. *Information and Control*, 56:154–173, 1983.

[MR86] H. Mannila and K.-J. Räihä. Inclusion dependencies in database design. In *Proceedings of the International Conference on Data Engineering*, pages 713–718, Los Angeles, Ca., 1986.

[MR92] H. Mannila and K.-J. Räihä. *The Design of Relational Databases.* Addison-Wesley, Reading, MA, 1992.

[Sci86] E. Sciore. Comparing the universal instance and relational data models. In P.C. Kanellakis and F. Preparata, editors, *Advances in Computing Research*, volume 3, pages 139–162. JAI Press, Greenwich, 1986.

[Ull88] J.D. Ullman. *Principles of Database and Knowledge-Base Systems*, volume 1. Computer Science Press, Rockville, MD, 1988.

[Vin92] M.W. Vincent. Modification anomalies and Boyce-Codd normal form. In B. Srinivasan and J. Zeleznikow, editors, *Research and Practical Issues in Data Management*, pages 251–264. World Scientific Press, Singapore, 1992.

[Vin94] M.W. Vincent. *The Semantic Justification for Normal Forms in Relational Database Design.* PhD thesis, Department of Computer Science, Monash University, Melbourne, 1994.

[Vin98] M.W. Vincent. Redudnancy elimination and a new normal form for relational databases. In B. Thalheim and L. Libkin, editors, *Semantics in Databases*, pages 247–264, Berlin, 1998. Springer Verlag.