1 Introduction

Goals:

1. Formalize the following three database design problems:
   a. Attribute redundancy - through the Redundancy Free Normal Form (RFNF)
   b. Violation of entity integrity – through generalized entity integrity
   c. Interaction between Functional Dependencies (FDs) and Inclusion Dependencies (INDs) through the non-interaction implication problem for FDs and INDs
2. Provide sufficient and necessary semantics for Inclusion Dependency Normal Form (IDNF).
3. Justify IDNF as a robust normal form that eliminates the above-mentioned three problems.

---

2 Notation

U	A finite set of attributes
R, R_i	A relational schema – a finite sequence of distinct attributes from U
R	A database schema
D	A countably infinite domain of values, linearly ordered
t, t_i	An R-tuple / tuple in r, r_i
r, r_i	A relation over R, R_i
d	A database over **R**
X,Y,Z	A set of attributes
A,B,C	Attributes
t[Y]	The projection of t onto Y, the restriction of t to Y
$\Pi_Y(r)$	The projection of r onto Y, given by $\{ t[Y] \mid t \in r \}$
XY	X U Y (X and Y are sets)
F_i	And FD over R_i
I_i	An IND over R_i
σ	An FD or an IND
F	A set of FDs over **R**
I	A set of INDs over **R**
F[Y]	The projection of F onto Y over R, given by: $\{ R: X \rightarrow Z \mid R: X \rightarrow Z \in F^+ \text{ and } XZ \subseteq Y \}$
Σ	F U I (union of all FDs and INDs)
Σ^+	All the FDs and INDs that are logically implied by Σ (transitive closure)

3 Definitions

Definition 3.1: **Functional Dependency (FD)**

And FD over a relational schema R is a statement of the form R: X→ Y, where $X,Y \subseteq R$

- Trivial FD: If $Y \subseteq X$
- Standard FD: If $X \neq \varnothing$
- $d \models R: X \rightarrow Y$: If for each $t_1,t_2 \in r$: $t_1[X] = t_2[X] \rightarrow t_1[Y] = t_2[Y]$

Definition 3.2: **Inclusion Dependency (IND)**

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**, $X \subseteq R_i$, $Y \subseteq R_j$ and X,Y are sets of distinct attributes such that $|X| = |Y|$

- Trivial IND: Of the form $R[X] \subseteq R[X]$
- Unary IND: If $|X| = 1$
- Typed IND: If X=Y
- $d \models R_i[X] \subseteq R_j[Y]$: If $\Pi_X(r_i) \subseteq \Pi_Y(r_j)$ where r_i, $r_j \in d$ are relations over R_i and R_j respectively

Illustration:

R_1 = HEAD = {H,D}   (H – Head of department, D – Department)
R_2 = LECT = {L,D}   (L – Lecturer, D – Department)

FDs stating that a head of a department manages a unique department and a lecturer works in a unique department:

F = {HEAD: H→D, LECT: L→D}

IND stating that a head of a department also works as a lecturer in the same department:

I = {HEAD[HD] $\subseteq$ LECT[LD]}

Definition 3.3: **Superkey**

A set of attributes $X \subseteq R_i$ is a superkey for R_i with respect to FD F_i if

$F_i \models R_i : X \rightarrow R_i$ holds.

Definition 3.4: **Key**

If X is a superkey and for no $Y \subset X$, Y is a superkey, then X is a key.

Definition 3.5: **KEYS(F)**

All FDs of the form $X \rightarrow R_i$, where X is a key for R_i

Definition 3.6: **Normal Forms** (Informal)

Normal Forms formalize the criteria for redundancy elimination.

Definition 3.7: **BCNF** (Boyce-Codd Normal Form)

A database schema **R** is in BCNF if for all $R_i \in$ **R**, for all non-trivial FDs $R_i : X \rightarrow Y \in F_i$, X is a superkey for R_i with respect to F_i.

BCNF Example:

$R[X_1,X_2,X_3]$ given $F = \{ X_2 \rightarrow X_3 \}$ is not in BCNF:

R		
X_1	X_2	X_3
1	a	b
2	a	b
3	a	b
4	c	d

Note: the basic concept behind BCNF if that each fact is represented only once. In the above example, the values (X_2, X_3) = (a,b) appear many times, for each distinct value of X_1. This redundancy is not only wasteful in terms of storage, but also causes anomalies in updates and deletions.

Definition 3.8: **Super-key based (Key based) IND**

$R_i[X] \subseteq R_j[Y]$ is superkey based (key based) if Y is a superkey (key) for R_j with respect to F_j

Definition 3.9: **Circular IND**

A set I of INDs over **R** is circular if one f the following holds:

1. IND $R[X] \subseteq R[Y] \in I$

or

2. $R_1[X_1] \subseteq R_2[Y_2]$, $R_2[X_2] \subseteq R_3[Y_3]$, … , $R_m[X_m] \subseteq R_1[Y_1]$ where the m > 1 relational schemas are distinct.

Definition 3.10: **Proper circular**

A set of INDs over **R** is proper circular if one of the following holds:

1. I is not circular (noncircular)

or

2. $R_1[X_1] \subseteq R_2[X_2] \subseteq R_3[X_3]$, … , $R_m[X_m] \subseteq R_1[X_1]$ where m > 1 relation schemas are distinct.

Definition 3.11: The **implication problem**

The implication problem is the problem of deciding whether $\sigma \in \Sigma^+$**.**

When we consider FDs and INDs together, the implication problem is, in general, undecidable / untraceable.

The Pullback Inference Rule

Allows inference of an FD from and FD and an IND.

If $\Sigma \models \{ R[XY] \subseteq S[WZ], S: W \rightarrow Z \}$

and $|X| = |W|$

then $\Sigma \models R: X \rightarrow Y$

Definition 3.12: **Reduced set** of FDs and INDs

- $Y \subseteq R_i$ is reduced with respect to F_i if $F_i[Y]$ contains only trivial FDs.
- Σ is reduced if for all $R_i[X] \subseteq R_j[Y] \in I$, Y is reduced with respect to F_j.

The CHASE Procedure for INDs

The CHASE procedure is an algorithm that tests and forces a database d to satisfy a set of FDs and INDs.

CHASE(d, Σ) is the result of applying the following rules to the current state of d as long as possible:

FD rule: If $R_j : X \rightarrow Y \in F_j$ and there exist $t_1, t_2 \in r_j : t_1[X] = t_2[X]$ and $t_1[Y] \neq t_2[Y]$

then: for all $A \in Y$, change all the occurrences in d of max $(t_1[A], t_2[A])$ to $\min(t_1[A], t_2[A])$

IND rule: If $R_i[X] \subseteq R_j[Y] \in I$ and there exists $t_i \in r_i : t_i[X] \notin \Pi_Y(r_j)$

then: add a tuple t_j over R_j to r_j, where $t_j[Y] = t_i[X]$, and for each $A \in R_j - Y$, $t_j[A]$ is assigned a new value greater than the greatest value in the current state of the database.

- If I is <u>proper circular</u>, then the CHASE procedure always terminates after a finite number of applications of the IND rule. In this case CHASE(d, Σ) $\models \Sigma$
- If I is <u>circular</u> then it does not always terminate. In this case, if after a finite number of applications of the IND rule resulting in d' we have: d' $\not\models$ F then it is said to violate F.

CHASE Rule Examples

FD Rule:

d = $\Sigma = \{X_1X_2 \rightarrow X_3\}$

R		
X_1	X_2	X_3
1	2	3
1	2	8
8	5	9

Applying the FD-rule once results in:

R		
X_1	X_2	X_3
1	2	3
1	2	**3**
3	5	9

IND Rule:

d = $\Sigma = \{ R_1[X_1X_2] \subseteq R_2[Y_1Y_3] \}$

R_1		
X_1	X_2	X_3
1	2	4
1	3	5

R_2		
Y_1	Y_2	Y_3
1	6	2

Applying the IND-rule once results in:

R_1		
X_1	X_2	X_3
1	2	4
1	3	5

R_2		
Y_1	Y_2	Y_3
1	6	2
1	**7**	**3**

4 Interaction between FDs and INDs

A set of FDs F over **R** is said not to interact with a set of INDs I over **R** if:

1. For all FDs α over **R**, for all subsets $G \subseteq F$, $G \cup I \models \alpha$ iff $G \models \alpha$

and

2. For all INDs β over **R**, for all subsets $J \subseteq I$, $F \cup J \models \beta$ iff $J \models \beta$

Interaction between FDs and INDs is undesirable for two reasons:

1. The interaction may lead to the logical implication of data dependencies that were not envisaged by the database designer and may be very difficult to find.
2. A database design may be normalized with respect to the set of FDs but not with respect to the combined set of FDs and INDs.

Example Illustrating the Second Problem:

R = {R,S}, R = S = ABC, F = { S: A→ BC } I = { R[AB] $\subseteq$ S[AB] }

R is in BCNF with respect to F.

R: A→B is logically implied by Σ (pullback inference rule)

Now if we add R:A→B to F, then **R** is not in BCNF since A is not a superkey for R with respect to {R:A→B}

Theorem:

If:

1. **R** is in BCNF with respect to F over **R**
2. I is a proper circular set of INDs over **R**
3. Σ is reduced

then:

F and I do not interact.

5 Attribute Redundancy

Definition 5.1: **Value Redundancy**

The occurrence of a value t[A] is redundant in d with respect to F if for every replacement of t[A] by v ≠ t[A], v ∈ D, resulting in database d' we have that d' ⊭ F.

Definition 5.2: **VRFNF Value Redundancy Free Normal Form**

If there does not exist a database d over **R** with redundant values, **R** is said to be in VRFNF form.

Theorem

R is in BCNF iff **R** is in VRFNF

Attribute Redundancy Example

R		
A	B	C
1	2	3
1	2	4

F = { R: A→ B}

R is not in VRFNF:

R		
A	B	C
1	2	3
1	**3**	4

(F is now violated)

Lemma

Let Σ be a set of FDs and noncircular INDs over a database schema **R**.

Then, **R** is not in VRFNF with respect to Σ if I contains at least one nontrivial IND.

Proof

$I_i = R_i[X] \subseteq R_j[Y]$ (non-trivial)

d:

R		
A	B	…
0	0	…

all other relations are empty

d' = CHASE (d, Σ) $\Rightarrow$ d' $\models \Sigma$

I is not circular $\Rightarrow r_i = r_i'$

Y values in r_j' must contain zeros because d' $\models \Sigma$

d':

	R_i		
	A_i	B_i	…
t_i	0	0	…

R_j		
A_j	B_j	…
0	0	…

all other relations are empty

The zeros $t_i[A_i]$, $t_i[B_i]$… are redundant since changing any of them violates I_i.

Definition 5.3: **Attribute redundancy**

Let d be a database over **R** which satisfies Σ, and $r \in d$ a non-empty relation over R. If for every tuple $t \in r$, t[A] replaced by a distinct value $v \in D$, v ? t[A], results in the database d', $d' \not\models \Sigma$, then A is said to be redundant.

Definition 5.4**: ARFNF** – **Attribute Redundancy Free Normal Form**

R is said to be in ARFNF if there does not exist an attribute A in $R \in$ **R** which is redundant with respect to Σ.

Example:

$R_1 = \{H,D\}$

$R_2 = \{L,D\}$

$F = \{R_2: L \rightarrow D\}$

$I = \{R_1[HD] \subseteq R_2[LD]\}$

$\Rightarrow$ The attribute D in R_1 is redundant (attribute redundancy).

Definition 5.5: **RFNF Redundancy Free Normal Form**

R is said to be in RFNF with respect to Σ over **R** if it is in VRFNF with respect to F and in ARFNF with respect to Σ.

6 Update Anomalies

Three types: Insertion, deletion, modification.

Observation: Let r and s be relations over R such that $s \subseteq r$ then:

$$r \models F \Rightarrow s \models F$$

Conclusion: There are no deletion anomalies for FDs.

We will restrict our study in Insertion and modification anomalies.

Definition 6.1 Compatible tuple

t is compatible with d if d U {t} $\models$ KEYS(F)

Definition 6.2 Insertion violation

A database d has an insertion violation if:

1. d $\models \Sigma$

   and

2. there exists t compatible with d: CHASE(d U {t},**I**) $\not\models \Sigma$

Definition 6.3 Free of insertion anomalies

A database schema **R** is free of insertion anomalies if there does not exist d over **R** which has an insertion violation.

Theorem

For I not circular:

R is free of insertion anomalies iff Σ is reduced and **R** is in BCNF with respect to F.

Insertion Violation Example 1

R = {HEAD,LECT}   HEAD = {H,D}   LECT = {L,D}

Σ = {HEAD: H→D, LECT: L→D, HEAD[HD] $\subseteq$ LECT[LD]}

Note: Σ is not reduced. F_{LECT}[LD] includes a nontrivial FD: LECT: L→D

R is in BCNF – H is key in HEAD, and L is key in LECT.

r_1 over HEAD is empty.

r_2 =

LECT	
L	D
Jane	CS

t = <Jane, Math> over r_1 is compatible with d.

We insert t into r_1 and apply the CHASE procedure:

r_1' =

HEAD	
H	D
Jane	Math

r_2' =

LECT	
L	D
Jane	CS
Jane	Math

r_2 violates LECT L→D

Insertion Violation Example 2

<u>R</u> is not in BCNF

<u>R</u> = {R}

R = {A,B,C}

Σ = { R: A→B}

Note: Σ is reduced, **<u>R</u>** is not in BCNF – A is not a key in R.

d = r =

R		
A	B	C
a_1	b_1	c_1

t = <a_1,b_2,c_2> over r is compatible with d.

We insert t into r and get:

d’ = r’ =

R		
A	B	C
a_1	b_1	c_1
a_1	b_1	c_2

r_2’ violates r: A→B

Modification Anomalies

Definition 6.3 Modification Violation

d over **R** has a modification violation with respect to Σ over **R** if:

1. $d \models \Sigma$

and

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

Definition 6.4 Free of modification anomalies

R is free of modification anomalies with respect to Σ if there does not exist a database d over **R** with a modification violation.

Theorem:

Let Σ be a set of FDs and noncircular INDs over **R**. **R** is free of modification anomalies iff **R** is free of insertion anomalies.

Modification violation example

R = {HEAD,LECT} HEAD = {H,D} LECT = {L,D}

Σ = {HEAD:H→D, LECT:L→D, HEAD[HD] $\subseteq$ LECT[LD]}

HEAD	
H	D
Carmit	Math

LECT	
L	D
Carmit	Math

t = <Carmit,Physics>

We replace <Carmit,Math> with t in LECT and get:

HEAD	
H	D
Carmit	Math

LECT	
L	D
Carmit	Physics

Applying CHASE we get a violation of H→D

7 Generalized Entity Integrity

To illustrate the problem of entity integrity violation we consider the following example, in which propagation of insertions due to INDs results in the violation of entity integrity:

R = {EMP,PROJ}
EMP = {Employee, Project}, PROJ = {Project, Location}
Σ = {EMP: Employee $\rightarrow$ Project, EMP[Project] $\subseteq$ PROJ[Project]}
r_1 is over EMP, r_2 is over PROJ
t = <c,p> over r_1 where p is a new project which has not yet been allocated a location and thus is not yet recorded in r_2.
We insert t into r_1. Due to the IND, the insertion should be propagated to r_2 by inserting into r_2 a tuple recording the new project. However, 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.

Proposition:
Superkey-based INDs do not cause the propagation of the insertion of tuples that represent undefined entities.

Let t be a tuple being added to r_i in the current state of d during the computation of CHASE (d, Σ).
t is **entity based** if for each X key for R_i , for all A in X: t[A] is not a new value assigned to t as a result of applying the IND rule.
R satisfies **generalized entity integrity** with respect to Σ if for each d over **R**, all the tuples that are added to d in its current state during the computation of CHASE (d, Σ) are entity-based.

Theorem
R satisfies generalized entity integrity with respect to Σ iff I is superkey-based.

8 Inclusion Dependency Normal Form

Definition: **Inclusion Dependency Normal Form**

R is in IDNF with respect to ∑ if:

1. **R** is in BCNF with respect to F

and

2. I is not circular and key-based.

Special Case: If I = ∅ then **R** being in IDNF is equivalent to it being in BCNF.

Theorem

For I not circular:

The following statements are equivalent:

1. **R** is in IDNF
2. **R** is free of insertion anomalies and satisfies generalized entity integrity
3. **R** is free of modification anomalies and satisfies generalized entity integrity
4. **R** is in RFNF and satisfies generalized entity integrity.

References

1. M.Levene and M.W.Vincent, "Justification for Inclusion Dependency Normal Form", *IEEE Transactions on Knowledge and Data Engineering,* vol.12, no.2, March / April 2000.
2. Makowsky, "Lecture #10 – Normal Forms" – *Database Management Systems* Course, Technion.
3. Makowsky, "Lecture #11 – Why Boyce-Codd Normal Form?" – *Database Management Systems* Course, Technion.