BCNF revisited: 30 Years Normal Forms

J.A. Makowsky
Faculty of Computer Science
Technion - IIT, Haifa

janos@cs.technion.ac.il
www.cs.technion.ac.il/~janos

Based on work by **M.W. Vincent**,
Advanced Computing Research Center,
University of Southern Australia, Adelaide, Australia 5095,
e-mail: millist.vincent@unisa.edu.au

and joint work with **E.V. Ravve**,
Israel Design Center, Intel Corporation, Haifa, Israel
e-mail: elena.ravve@intel.com

Overview

- Normal forms and functional dependencies
- BCNF and redundancy
- BCNF and update anomalies
- BCNF and storage saving
- Achieving BCNF
- Other normal forms

Functional Dependencies

$U = \{A_1, A_2, \ldots, A_m\}$ a set of attributes

F a set of functional dependencies for $R[U]$ of the form $X \to Y$ with $X, Y \subseteq U$.

F^+ the deductive closure of F
(with respect to the Armstrong axioms).

$K \subseteq U$ is a **superkey** for F if $K \to U \in F^+$.
$K \subseteq U$ is a **key** for F if K is a superkey, but no $K' \subset K$ is a superkey.

$F_{key} = \{K \to U \in F^+ : K \text{ is a key }\}$.

Normal Forms

$(R[U], F)$ is in **Boyce-Codd Normal Form** or

$(R[U], F)$ is in **BCNF**

if $F^+_{Key} = F^+$.

$(R[U], F)$ is in **Third Normal Form** or

$(R[U], F)$ is in **3NF**

if for every non-trivial $X \to Y \in F^+$ either

- X is a superkey or
- $Y \subset K$ for some key K for F.
  This is called a BCNF-violation for the key K

Examples for Normal Forms

The relation scheme $R[CSZ]$ with
C City
S Street
Z Zipcode
and $CS \to Z, Z \to C$ is in 3NF but not in BCNF.

CS is the only key
$Z \to C$ is a BCNF-violation.

Examples for Normal Forms, II

The relation scheme $R[NSCAP]$ with
N Name
S Street
C City
A Areacode
P Phone number
and $NSC \to AP$, $SC \to A$, is not in 3NF.

NSC is the only key

$R_1[NSCP]$ with $NSC \to AP$, and
$R_2[SCA]$ with $SC \to A$,
are both in BCNF.

Why Boyce Codd Normal Form ?

- They avoid redundancy
- They avoid update anomalies
- They minimize storage

We have to make this precise.

Redundancy, I

Let R, F be a relation scheme.

R is *F-redundant (F^+-redundant) on XY*
if there exists a relation $r \models F$
and a non-trivial FD $X \to Y \in F$ ($\in F^+$),
and at least two distinct tuples $t_1, t_2 \in r$
with $t_1[XY] = t_2[XY]$.

R with $F = \{A \to B, BC \to A\}$ is F-redundant,
and hence F^+-redundant.

	R	
A	B	C
a_1	b_1	c_1
a_1	b_1	c_2

Redundancy, II

The set of attributes of the form XY

- with $X \rightarrow Y \in F$ and not trivial, are called *explicit facts*.
- with $X \rightarrow Y \in F^+$ and not trivial, are called *implicit facts*.

The rationale behind redundancy is, that if R is redundant on a fact, the fact should be stored in an different table.

Redundancy, III

Theorem:
(Bernstein, Goodman, 1980; M.W. Vincent 1994)

The following are equivalent:

(i) R, F is in BCNF;

(ii) R, F is not F-redundant;

(iii) R, F is not F^+-redundant;

Insertion anomalies, I

We are given a relation scheme $R[U]$ and a set of FD's F with a set of candidate keys given by F_{Key}.

Let r be a relation for R with $r \models F$.

Let $t[U]$ be a tuple we want to insert.

We check whether $r \cup \{t[U]\} \models F_{Key}$.

If $r \cup \{t[U]\} \models F_{Key}$ we accept, else we reject the insertion of $t[U]$.

If we accept, but $r \cup \{t[U]\} \not\models F$, we say that $t[U]$ is an *insertion violation*, IV.

R, F has an *insertion anomaly* if there is an r and $t[U]$, which is an insertion violation.

Insertion anomalies, Example

We look at $R[A, B, C]$ with
$F = \{A \rightarrow B, B \rightarrow C\}$.

	R	
A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

We want to insert (a_3, b_1, c_3).

This is compatible with $F_{Key} = \{A \rightarrow BC\}$.

	R	
A	B	C
a_1	b_1	c_1
a_2	b_2	c_2
a_3	b_1	c_3

But this violates $B \rightarrow C$.

Insertion anomalies, Theorem

Recall R, F is in BCNF iff $F_{Key} \models F$.

Theorem: (R. Fagin, 1979)
R, F is in BCNF iff
it has no insertion anomalies.

Proof:
Assume $F_{Key} \models F$, $r \models F$ and $r \cup \{t\} \models F_{Key}$.
Then $r \cup \{t\} \models F$.

The other direction needs some work.

Deletion anomalies, I

We are given a relation scheme $R[U]$ and a set of FD's F with a set of candidate keys given by F_{Key}.

Let r be a relation for R with $r \models F$.

Let $t[U] \in r$ be a tuple we want to delete.

We check whether $r - \{t[U]\} \models F_{Key}$.

If $r - \{t[U]\} \models F_{Key}$ we accept, else we reject the deletion of $t[U]$.

If we accept, but $r - \{t[U]\} \not\models F$, we say that $t[U]$ is an *deletion violation*, DV.

R, F has an *deletion anomaly* if there is an r and $t[U]$, which is an deletion violation.

Deletion anomalies, II

Observation:
Let r be a relation for R and F a set of FD's.
Let $s \subseteq r$ another relation for R.

If $r \models F$ so also $s \models F$.

Conclusion:
There are no deletion anomalies for FD's.

Note: In the presence of Multivalued Dependencies (MVD's) there may occur deletion anomalies.

Modification anomalies, I

Let r be a relation for $R[U], F$, $t \in r$, $r \models F$, K_0 be a fixed candidate key for F.

Let t' be a tuple such that $(r - \{t\}) \cup \{t'\} \models F_{Key}$ and one of the following:

(i) $t[K] = t'[K]$ for some candidate key for F;

(ii) $t[K_0] = t'[K_0]$;

(iii) $t[K] = t'[K]$ for every candidate key for F;

but $(r - \{t\}) \cup \{t'\} \not\models F$

Then r and t' show a *modification anomaly* M_i, M_{ii}, M_{iii} respectively.

Modification anomalies, Example

$R[ABC]$ with $F = \{A \rightarrow B, BC \rightarrow A\}$
Candidate keys AC, BC. Choose $K_0 = BC$.

	A	B	C
	a_1	b_1	c_1
t=	a_1	b_1	c_2
s=	a_2	b_2	c_2

We modify once t and once s:

	A	B	C
	a_1	b_1	c_1
t'=	a_1	b_2	c_2

$t[AC] = t'[AC]$ and F_{Key} is satisfied,
but $A \rightarrow B$ is violated.

	A	B	C
	a_1	b_1	c_1
s'=	a_1	b_2	c_2

$s[BC] = s'[BC]$ and F_{Key} is satisfied,
but $A \rightarrow B$ is violated.

In this example we cannot take care of both candidate keys simultaneously.

Modification anomalies, II

Clearly, every M_{iii} anomaly is also an M_{ii} anomaly, and every M_{ii} anomaly is also an M_i anomaly.

Observation:

If R, F is in BCNF then it has no modification anomaly M_i (and hence neither M_{ii} and M_{iii}).

Proof: Use that $F_{key} \models F$.

Modification anomalies, III

Theorem:(M.W. Vincent, 1994)

The following are equivalent:

(i) R, F is in BCNF

(ii) R, F has no modification anomaly M_i

(iii) R, F has no modification anomaly M_{ii}

Vincent also introduces a normal form weaker than BCNF, but stronger than 3NF which is characterized by the absence of M_{iii} modification anomalies.

Unpredictable insertions, I

Let $R[U], F$ be a relation scheme.
An insertion of a tuple t into $r \models F$ is said to be F-*valid*, if $r \cup \{t\} \models F$.

A set of attributes $X \subseteq U$ is said to be *unaffected* by a valid insertion $r' = r \cup \{t\}$ iff $\pi_X(r) = \pi_X(r')$.

A valid insertion is F-unpredictable
(F^+-unpredictable)
if there exists a non-trivial $X \rightarrow Y \in F$
($X \rightarrow Y \in F^+$)
such that XY is unaffected by it.

Unpredictable insertions, Example

$R[ABC]$ with $F = \{A \rightarrow B, BC \rightarrow A\}$
We look at $A \rightarrow B$:

	A	B	C
	a_1	b_1	c_1

We now insert t

	A	B	C
	a_1	b_1	c_1
t=	a_1	b_1	c_2

This is a valid insertion which does not affect AB. Hence it is F-unpredictable.

Clearly, F-unpredictable implies F^+-unpredictable.

Unpredictable insertions, II

Observation:

If R, F has a F^+-unpredictable insertion, then it is not in BCNF.

Proof:
There is r and t such that $r \cup \{t\} \models F$
and hence $r \cup \{t\} \models F_{Key}$.

There is some non-trivial $X \rightarrow Y \in F^+$, and $t' \in r$ with $t \neq t'$ but $t[XY] = t'[XY]$.

Assume for contradiction, R, F is in BCNF.
So X is a superkey for F.
But $r \cup \{t\} \models F_{Key}$.
So $t = t'$, a contradiction.

Unpredictable insertions, III

Theorem: (Bernstein, Goodman, 1980)

The following are equivalent:

(i) R, F is in BCNF;

(ii) R, F has no F-unpredictable insertions.

(iii) R, F has no F^+-unpredictable insertions.

Minimizing storage, I

Let $R[U], F$ be a relation scheme,
and $\pi_{U_i} R = R_i[U_i]$ be a
information preserving decomposition,
i.e. $F \models \bowtie_i R_i[U_i] = R$.

We say that the decomposition is
storage saving if there are instances $r = \bowtie_i r_i$
such that $\sum_i |r_i| \leq |r|$.

Example:

Consider $R[ABCD]$ with
$F_1 = \{A \rightarrow BCD, C \rightarrow D\}$ (not in BCNF) and
$F_2 = \{A \rightarrow BCD, C \rightarrow A\}$ (in BCNF) and

We decompose R into
$R_1[ABC]$ and $R_2[CD]$ for F_1
and $S_1[AC]$ and $S_2[ABD]$ for F_2.

With F_1 there may be fewer values for C than for A, but with F_2 this is not possible.

Minimizing storage, II

Observation:

If R, F is in BCNF then it has no storage saving decomposition.

Theorem:(Biskup; Vincent and Srinivasan)

If R, F is in BCNF iff it has no storage saving decomposition.

Remark: This holds also for wider dependency classes and their respective normal forms.

Splitting zip-codes, I

The examply $R[CSZ]$ with
C: City, S: Street, Z: Zipcode
and $CS \rightarrow Z$, $Z \rightarrow C$ is in 3NF
but not in BCNF.
The only BCNF-violation is $Z \rightarrow C$.

We can bring it into BCNF in two ways:

- Drop $Z \rightarrow C$

  The character of postal distribution has changed

- Split Z into Z_{city} and Z_{local} with
  $CS \rightarrow Z_{local}$, $Z_{city} \rightarrow C$, $C \rightarrow Z_{city}$
  and new relations
  $S_1[CSZ_{local}]$ and $S_2[C, Z_{city}]$.

  Many countries do this

Splitting zip-codes, II

We split the zip-code Z into Z_{City} and Z_{local} and store it more efficiently:

$ZipCode[SZ_{City}Z_{local}]$ with $Z_{City}S \rightarrow Z_{local}$
the zip-code table and

$CityCode[CZ_{City}]$ with $C \leftrightarrow Z_{City}$
the city-zip-code table.

We have two tables instead of one.
But we can gain storage space provided

- Z_{City} is a short code for city names, and
- Z_{local} is a short code for sets of street names.

Note that saving storage must be measured in bits not in the number of tuples.

Splitting zip-codes, III

If we drop the BCNF-violation from our requirements, we save even more storage:

We can use the unused zip-codes resulting from inbalances of city-size:

- New York has many zip-codes,
  say 001-0001 up to 001-9999
- Montauk has very few,
  say 002-0001 up to 002-0009
- With $Z \rightarrow C$ the values
  002-0010 up to 002-9999 are waisted.
- We can also gain by grouping small cities into bigger areas with same first three digits.

Hidden Bijections

Let $R[VXY], F$ be a relation scheme with V, X, Y disjoint sets of attributes and F a set of FD's.

We say that F has a *hidden bijection* if

$$VX \leftrightarrow VY \in F^+$$

and

$$Y \rightarrow X \in F^+ \text{ or } X \rightarrow Y \in F^+$$

The rôles of X and Y are **not** symmetric.

Proposition:(M.-Ravve)
$(R[U], F)$ is in BCNF iff
it has no hidden bijections.

Attribute splitting, I

Let $R[VXY], F$ be a relation scheme with V, X, Y disjoint sets of attributes and F a set of FD's,
and $VX \rightarrow VY$ and $Y \rightarrow X$ in F^+ a hidden bijection.

For $A \in Y$ an VX*-splitting of* A *into* A_V, A_X is given by

$R_1[VA_XA_V(Y-A)]$
with $VA_X \rightarrow A_V$ and $VA_X \rightarrow (Y-A)$,
$R_2[XA_X(Y-A)]$
with $A_X(Y-A) \leftrightarrow X$,
$R_3[A_XA_VA]$
with $A_VA_X \leftrightarrow A$.

Attribute splitting, II

Conversely, given

$R_1[VA_XA_V(Y-A)]$, $R_2[XA_X(Y-A)]$, $R_3[A_XA_VA]$
with $VA_X \rightarrow A_V(Y-A)$, $A_X(Y-A) \leftrightarrow X$, and $A_VA_X \leftrightarrow A$,

we form first $S_1 = R_1 \bowtie R_2$ and then S_2
by fusing in S_1 A_1A_2 into A (using R_3) .

If S_2 has the same instances as R, we say the *attribute splitting is information preserving*.

It follows that $S[VXY]$ $VX \rightarrow Y$ holds
and also, either $Y \rightarrow X$ or $Y \rightarrow V$.

Proposition:(M.-Ravve, 2002)
If attribute splitting
in $(R[VXY], F)$ is information preserving,
then F has a hidden bijection.

Attribute splitting and storage saving

R

X	V	A	Y-A

becomes

R_1

V	A_X	A_V	Y-A

R_2

X	A_X	Y-A

R_3

A_X	A_V	A

Observation: For every $A \in Y$ there are instances of R for which the VX-splitting of A is storage saving (in bits).

BCNF and splittings

Proposition:(M.-Ravve 2002)

A relation scheme (R, F) is in BCNF iff it allows no storage saving via information preserving attribute splitting.

Proof:

If (R, F) allows information preserving attribute splitting it must have a hidden bijection (by the previous proposition).

But we have seen that (R, F) is in BCNF iff it has no hidden bijections.

Can we achieve BCNF ?

It is well known that

there are relation schemes $R[U], F$

- which are not in BCNF and
- do not allow
  information preserving and
  dependency preserving
  decomposition via *projections*.

Achieving Normal Forms

- Using projection-decompositions only
  we can get BCNF
  but cannot guarantee the dependencies.
- Using synthesis algorithms
  we can get 3NF
  but cannot always avoid hidden bijections.
- We shall combine
  - projection-decompositions
  - synthesis, and
  - attribute splitting.

Another example

We now look at the examply $R[ABCSZ]$ with $F = \{CS \to Z, Z \to C, B \to C, ZA \to B\}$.

The keys are CSA, BSA, ZSA.
$R[ABCSZ]$ is in 3NF but not in BCNF.
All FD's in F are BCNF violations.
F is a minimal cover.

Synthesis gives
$R_1[CSZ]$, $R_2[BC]$, $R_3[ABZ]$ and $R_{Key}[CSA]$ with
$F_1 = \{CS \to Z, Z \to C\}$,
$F_2 = \{B \to C\}$,
$F_3 = \{ZA \to B\}$ and $F_{Key} = \emptyset$.

Another example (continued)

$R_1[CSZ]$, $R_2[BC]$, $R_3[ABZ]$ and$R_K[CSA]$
with $F = \{CS \to Z, Z \to C, B \to C, ZA \to B\}$.

We split Z into Z_S, Z_C for R_1 and $Z \to C$.
We replace R_1 by $S_1[CSZ_S]$ with key CS.
We add $S_2[CZ_C]$ with $C \leftrightarrow Z_C$.

What do we do in $R_3[ABZ]$?

(Bad) We replace it by
$S_3[ABZ_SZ_C]$ with key Z_SZ_CA.
But this has a new
BCNF-violation $B \to Z_C$.

(Good) We leave $R_3[ABZ]$
but add a new relation
$S_4[ZZ_SZ_C]$ with $Z \leftrightarrow Z_SZ_C$.

Splitting in minimal covers, I

Let F be a minimal cover for $R[U]$ and
$X \to A \in F$.

Assume:
Synthesis gives an $S[XA]$ with F_1 a minimal cover (derived from F).

Assume:
X is the only key of $S[XA]$ (via F_1).

A BCNF-violation for $S[XA]$ for the key X is of the form
$AY_1 \to B_1$ with $Y_1 \subset X$, possibly empty, and $B_1 \in X - Y_1$.

As AY_1 is not a superkey for $S[AX]$,
$Y_1B_1 \subset X$ is a proper subset.

Splitting in minimal covers, II

Assume X is the only key of $S[XA]$ (via F_1).

Let the BCNF-violations for X be
$AY_i \to B_i$, $i \geq 1$.

We split A and get $S_1[XA_{Y_1}]$, $T_1[A_{B_1}B_1]$ and $T_1^*[AA_{B_1}A_{Y_1}]$.

Put $\hat{F}_1 = (F_1 - \{X \to A, A_1Y_1 \to B_1\})$
$\bar{F}_1 = \{X \to A_{Y_1}, A_{B_1} \leftrightarrow B_1, A \leftrightarrow A_{B_1}A_{Y_1}\}$
$F_{split(A)} = \hat{F}_1 \cup \bar{F}_1$

Claim:
(i) $F_{split(A)}$ is a minimal cover for $F_{split(A)}$ and the relations S_1, T_1, T_1^*.
(ii) $\bar{F}_1$ has no BCNF-violations.
(iii) $\hat{F}_1$ has fewer BCNF-violations than F_1

Splitting in minimal covers, III

(i) $F_{split(A)}$ is a minimal cover for $F_{split(A)}$ and the relations S_1, T_1, T_1^*.

Proof: Use that $(F_1 - \{X \to A, Y_1 \to B_1\})$ is a minimal cover, because it is a subset of a minimal cover and does not contain the split attributes.
The new FD's do not create any
new redundancies.

(ii) $\bar{F}_1$ has no BCNF-violations.

Proof: Inspect
$\bar{F}_1 = \{X \to A_{Y_1}, A_{B_1} \leftrightarrow B_1, A \leftrightarrow A_{B_1}A_{Y_1}\}$
and the relations S_1, T_1, T_1^*.

(iii) $\hat{F}_1$ has fewer BCNF-violations than F_1

Proof: Use $\hat{F}_1 \subset F_1$.

Splitting in minimal covers, IV

We have not yet reached the general situation:

(i) A was assumed to be a singel attribute, but it could be a set $\{A_1, \ldots, A_m\}$.

Split all the A_i's simulatenously into $A^i_{B_1}$ and $A^i_{Y_1}$.

(ii) There could be some other key K for $S[XA_1, \ldots, A_m]$ and a BCNF-violation for K of the form $A_1, \ldots A_k Y_1 \to B_1$ with $k \leq m$ and $B_1 \in K$.

Put $U = XA_1, \ldots A_m$.
Find a new minimal cover for F_1 which contains $K \to U_K$ and $A_1, \ldots A_k Y_1 \to B_1$.
Write $A_1, \ldots A_k Y_1 = V_1 {Y_1}'$ with $V_1 \subset U - K$ and ${Y_1}' \subset K$.

BCNF via splitting attributes, II

Theorem: (Makowsky, Ravve 1998, 2002)

Every relation scheme R, F can be modified preserving information and dependencies via decomposition and splitting attributes.

Furthermore, this modification can be computed using a combination of the synthesis algorithm for 3NF and splitting attributes.

Inclusion Dependencies

Inclusion dependencies (IND's) are of the form

$$\pi_X R \subseteq \pi_Y S$$

where $X = (X_1, \ldots, X_m)$ $Y = (Y_1, \ldots, Y_m)$ and X_i and Y_i have the same domains.

An inclusion dependency $\pi_X R \subseteq \pi_Y S$ is

- *unary* iff $m = 1$;
- *key based* if Y is a key of S
- *superkey based* if Y is a superkey of S

Circularity of IND's

A set I of IND's for relationschemes R_i is *circular* if

- I contains a nontrivial $\pi_X R \subseteq \pi_Y R$, or
- there exists relation schemes $R_{j_1}, \ldots, R_{j_m}$ such that I contains

$$\pi_{X_{j_1}} R_{j_1} \subseteq \pi_{X_{j_2}} R_{j_2} \subseteq \ldots \subseteq \pi_{X_{j_m}} R_{j_m} \subseteq \pi_{X_{j_1}} R_{j_1}$$

We note that circularity is a syntactic property, hence decidable.

Consequence problem for IND's

(i) (Casanova, Fagin, Papadimitriou, 1984)
The consequence problem for IND's alone is decidable (in fact PSpace-complete).

(ii) (Mitchell 1983, Chandra and Vardi 1985)
The consequence problem for IND's with FD's undecidable.

(iii) (Cosmodakis, Kannelakis, 1986)
The consequence problem for non-circular IND's with FD's is decidable (in fact ExpTime-complete).

(iv) (Cosmodakis, Kannelakis and Vardi, 1990)
The consequence problem for unary IND's with FD's is decidable in polynomial time.

Inclusion Dependency Normal Form

after M. Levene and M.W. Vincent

A set $F \cup I$ of FD's and non-circular IND's over a set of relationschemes $\mathbf{R} = (R_i)_{i \leq \ell}$ is in *Inclusion dependency normal form IDNF* if

- $\mathbf{R}, F$ is in BCNF
- I is key-based.

By the non-circularity assumption this is decidable.

Update anomalies for FD's and IND's

after M. Levene and M.W. Vincent

Insertion and modification anomalies can be defined similarly as for FD's alone.
However, there are some subtle points: The anomalies may occur only after the CHASE algorithm for IND's alone is applied.

Theorem:(Levene and Vincent, 2000)
The following are equivalent:

- $(\mathbf{R}, F \cup I)$ is in IDNF f
- $(\mathbf{R}, F \cup I)$ is free of insertion anomalies and superkey based.
- $(\mathbf{R}, F \cup I)$ is free of modifcation anomalies and superkey based.

Entity Integrity

after M. Levene and M.W. Vincent

Insertions and modifcations may propagated through several relations due to the IND's.

Levene and Vincent define a notion of

(Generalized) Entity Integrity (GEI)

which formalizes how this propagation should be kept under control.

Theorem:(Levene and Vincent, 2000)
A database scheme $(\mathbf{R}, F \cup I)$ satisfies GEI iff I is superkey based.

Previous work

Normalforms for FD's and IND's were first considered in the context of

Entity-Relationship design

M.A. Casanova and J.E. Amaral de Sa (1984)
J.A. Makowsky, V. Markowitz and U. Rotics (1986)
H. Mannila and K.-J. Räihä (1986)
V.A. Markowitz and J.A. Makowsky (1987, 1988)

H. Mannila and K.-J. Räihä
The design of relational databases
Addison Wesley, 1992

Further work:

T.-W. Ling and C.H. Goh (1992)
J. Biskup and P. Dublish (1993)

M. Levene and VM.W. Vincent (2000)
are the first to characterize these.

Future work

Here are some further challenges:

- What is the relationship between ER-normalform (ERNF)and IDNF?
  (ERNF was defined by Mannila and Räihä, 1993)
- Can we formulate information and dependency preserving decomposition and attribute splitting in the presence of IND's.
  Dependency preserving refinements of Makowsky and Ravve, 1998, may be useful here.
- Can we always achieve IDNF via decomposition and attribute splitting?

References

- H. Mannila and K.-J. Räihä,
  The design of relational databases,
  Addison Wesley, 1992
- M.W. Vincent,
  The sematic justification for normal forms in relational database design,
  PhD Thesis, Monash Uiversity, Melbourne, Australia, 1994
- J.A. Makowsky and E.V. Ravve,
  Dependency preserving refinements and the fundamental problem of database design,
  Data and Knowledge Engineering 24.3 (1998), pp. 277-312
- M. Levene and M.W. Vincent,
  Justification for inclusion dependency normal form,
  IEEE Transactions on Knowledge and Data Engineering, 12.2 (2000), pp. 281-291
- J.A. Makowsky and E.V. Ravve,
  BCNF revisited: 30 years normal forms.
  in preparation