COS221 - L23 - Normalisation - BCNF, 4NF and 5NF

Linda Marshall

26 April 2023

Overview of Normalisation

Table 14.1 Summary of Normal Forms Based on Primary Keys and Corresponding Normalization

Normal Form	Test	Remedy (Normalization)
First (1NF)	Relation should have no multivalued attributes or nested relations.	Form new relations for each multivalued attribute or nested relation.
Second (2NF)	For relations where primary key contains multiple attributes, no nonkey attribute should be functionally dependent on a part of the primary key.	Decompose and set up a new relation for each partial key with its dependent attribute(s). Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it.
Third (3NF)	Relation should not have a nonkey attribute functionally determined by another nonkey attribute (or by a set of nonkey attributes). That is, there should be no transitive dependency of a nonkey attribute on the primary key.	Decompose and set up a relation that includes the nonkey attribute(s) that functionally determine(s) other nonkey attribute(s).

Normalise to 3NF

1. R($\underline{a}$, b, c, R1($\underline{d}$, $\underline{e}$)) with FD's:
   a $\rightarrow$ {b, c}
   c $\rightarrow$ {d, e}
2. R($\underline{a}$, $\underline{b}$, c, d, e, {f}) with FD's:
   {a,b} $\rightarrow$ c
   a $\rightarrow$ d
   c $\rightarrow$ e
3. R($\underline{a}$, $\underline{b}$, c, d, e, {f}) with FD's:
   b $\rightarrow$ c
   a $\rightarrow$ d
   c $\rightarrow$ e

Overview of Normalisation

- A relation schema R is in 1NF if there are no multivalued attributes or nested relations.
- A relation schema R is in 2NF if every non-prime attribute A in R is not partially dependent on any key of R.
- A relation schema R is in 3NF if, whenever a nontrivial functional dependency $X \rightarrow A$ holds in R, either

  (a) X is a superkey of R, or

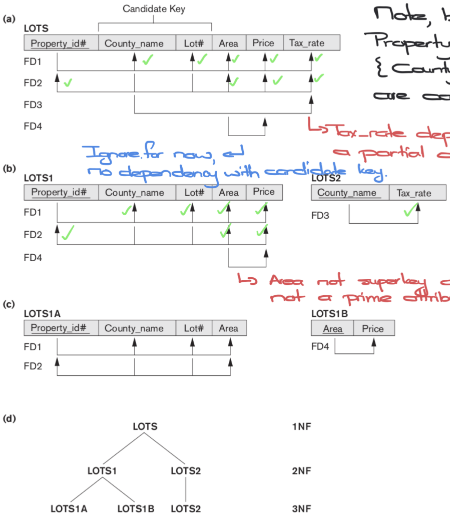
  (b) A is a prime attribute of R

  Violation of conditions (a) and (b) can occur when:
  - a nonprime attribute determines another nonprime attribute (typically resulting in a transitive dependency)
  - a proper subset of a key of R functionally determines a nonprime attribute (a partial dependency violates 3NF and 2NF)

  Another definition for 3NF - A relation schema is in 3NF if every nonprime attribute of R meets both the following conditions:
  - It is fully functionally dependent on every key of R
  - It is non-transitively dependent on every key of R

Overview

Figure 15.12
Normalization into 2NF and 3NF. (a) The LOTS relation with its functional dependencies FD1 through FD4. (b) Decomposing into the 2NF relations LOTS1 and LOTS2. (c) Decomposing LOTS1 into the 3NF relations LOTS1A and LOTS1B. (d) Summary of the progressive normalization of LOTS.



Note, both Property_id# and {County_name, Lot#} are candidate keys.

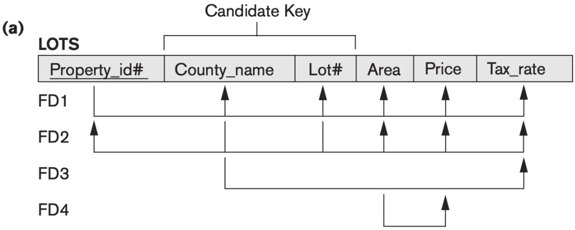
↳ Tax_rate dependent on a partial candidate key.

Ignore for now, as no dependency with candidate key.

↳ Area not superkey and Price is not a prime attribute.

Boyce-Codd Normal Form - BCNF

- BCNF was proposed to be simpler than 3NF, but is has been found to be stricter. Every relation in BCNF is also in 3NF.
- **Definition of BCNF**: A relation schema R is in BCNF if whenever a nontrivial functional dependency $X \rightarrow A$ holds in R, then X is the superkey of R.
- Consider the Lots example from before and assume the following functional dependency also exists:
  `FD5:  Area → County_name`



Boyce-Codd Normal Form - BCNF

- LOTS1A will then have the following FDs:

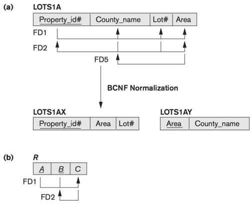


Figure 14.13
Boyce-Codd normal form. (a) BCNF normalization of LOTS1A with the functional dependency FD2 being lost in the decomposition. (b) A schematic relation with FDs; it is in 3NF, but not in BCNF due to the f.d. $C \rightarrow B$.

- LOTS1A is still in 3NF because County_name is a prime attribute. That is condition (b) of 3NF is satisfied.
- The relation is however not in BCNF. The condition (b) of 3NF is not part of the BCNF definition, only part (a) of the 3NF definition is.
- FD5 violates BCNF because Area is not a superkey of LOTS1A.
- LOTS1A is decomposed into two BCNF relations LOTS1AX and LOTS1AY. The functional dependency FD2 is lost because its attributes no longer co-exist in the same relation.

Fourth Normal Form - 4NF

- Sometimes relations have constraints that cannot be specified as FD's. An example of such a constraint is multivalued dependency (MVD). MVD's are a consequence of 1NF - an attribute in a tuple may not have a set of values.
- Consider the following example:

(a) **EMP**

Ename	Pname	Dname
Smith	X	John
Smith	Y	Anna
Smith	X	Anna
Smith	Y	John

- An employee with name `Ename` works on a project with name `Pname` and has a dependent with the name `Dname`.

Fourth Normal Form - 4NF

- Problem, an employee may work on many projects and have several dependents. To keep the relation state consistent and limit spurious relationships, a separate tuple to represent the employee-project and employee-project relationships must be defined.

(a) **EMP**

Ename	Pname	Dname
Smith	X	John
Smith	Y	Anna
Smith	X	Anna
Smith	Y	John

(b) **EMP_PROJECTS**

Ename	Pname
Smith	X
Smith	Y

EMP_DEPENDENTS

Ename	Dname
Smith	John
Smith	Anna

Fourth Normal Form - 4NF

- Informally, whenever two independent $1 : N$ relationships $A : B$ and $A : C$ are mixed in the same relation $R(A, B, C)$, an MVD may arise.
- Definition: $X \twoheadrightarrow Y$ (an MVD) specified on R, where X and Y are subsets of R, specifies the following constraint for $t_1[X] = t_2[X]$:
  - $t_3[X] = t_4[X] = t_1[X] = t_2[X]$
  - $t_3[Y] = t_1[Y]$ and $t_4[Y] = t_2[Y]$
  - $t_3[Z] = t_2[Z]$ and $t_4[Z] = t_1[Z]$, where Z is $(R - (X \cup Y))$

Fourth Normal Form - 4NF

- $X \twoheadrightarrow Y$ is trivial if:
  (a) - Y is a subset of X, or
  (b) - X U Y = R
- An MVD is nontrivial if neither (a) nor (b) holds.
- **Definition of 4NF**: A relational schema R is in 4NF with respect to its dependencies F (that is both FD's and MVD's), if for every nontrivial multivalued dependency $X \twoheadrightarrow Y$ in F' (all dependencies implied by F), X is the superkey for R.

Fourth Normal Form - 4NF

- A relation not in 4NF due to an MVD must be decomposed into separate relations (or trivial MVD).

(a) EMP

Ename	Pname	Dname
Smith	X	John
Smith	Y	Anna
Smith	X	Anna
Smith	Y	John

(b) EMP_PROJECTS

Ename	Pname
Smith	X
Smith	Y

EMP_DEPENDENTS

Ename	Dname
Smith	John
Smith	Anna

Fifth Normal Form - 5NF

- If no FD in R exists that violates any normal form up to BCNF, there may be no nontrivial MVD in R that violates 4NF. Enter the join dependency and 5NF.
- A join dependency (JD), denoted by $JD(R_1, R_2, ..., R_n)$ specified on relation R, specifies a constraint on the states r of R. This constraint states that every legal state, r of R should have a non-additive join decomposition into $R_1, R_2, ..., R_n$. For every r we have:

$$*(\pi_{R_1}(r), \pi_{R_2}(r), ..., \pi_{R_n}(r)) = r$$

- Note:
  - MVD is a two-dimensional (n = 2) version of JD.
  - Normalisation into 5NF is very rare.

Fifth Normal Form - 5NF

- **Definition: 5NF (also referred to as project-join normal form (PJNF))**: A relation schema R is in 5NF with respect to a set F of functional, multivalued and join dependencies, if, for every nontrivial join dependency $JD(R_1, R_2, ..., R_n)$ in F+ (that is implied by F), every R_i is a superkey of R.
- For example, consider the following relation with no MVDs which is in 4NF but not in 5NF because of a $JD(R_1, R_2, R_3)$ with R_1(Sname, Part_name), R_2(Sname, Proj_name) and R_3(Part_name, Proj_name).

(c) **SUPPLY**

Sname	Part_name	Proj_name
Smith	Bolt	ProjX
Smith	Nut	ProjY
Adamsky	Bolt	ProjY
Walton	Nut	ProjZ
Adamsky	Nail	ProjX
Adamsky	Bolt	ProjX
Smith	Bolt	ProjY

- Tuples below the dashed line must exist in any legal state of SUPPLY along with the tuples above the line.

Fifth Normal Form - 5NF

- Decomposition of the relation SUPPLY will result in the following relations.

(c) SUPPLY

Sname	Part_name	Proj_name
Smith	Bolt	ProjX
Smith	Nut	ProjY
Adamsky	Bolt	ProjY
Walton	Nut	ProjZ
Adamsky	Nail	ProjX
Adamsky	Bolt	ProjX
Smith	Bolt	ProjY

(d) R_1

Sname	Part_name
Smith	Bolt
Smith	Nut
Adamsky	Bolt
Walton	Nut
Adamsky	Nail

R_2

Sname	Proj_name
Smith	ProjX
Smith	ProjY
Adamsky	ProjY
Walton	ProjZ
Adamsky	ProjX

R_3

Part_name	Proj_name
Bolt	ProjX
Nut	ProjY
Bolt	ProjY
Nut	ProjZ
Nail	ProjX

- Note, by applying a natural join on 2 of the relations will result in spurious tuples. Applying a natural join on all 3 will not.

Summary

- A mechanism to check whether a relation schema is good or bad has been discussed.
- By properly designing the conceptual schema (ER and EER) and following the mapping of entities and relationships into relations and applying these guidelines will result in a sound database relation schema.
  - So, if you are in control of the entire database conceptual design, translation into relations and implementation in an RDBMS, you will not need to normalise.
  - If you inherit a database with no conceptual design and/or cannot figure out the mapping to the relation schema, you will need to test the goodness of the design by following the normalisation process.