Normalization - Relational Databases

Overview

- Normalization – An introduction
- Definition of Keys – Recap
- First Normal Form (1NF)
- Second Normal Form (2NF)
- Third Normal Form (3NF)
- Boyce-Codd Normal Form (BCNF)

Normalization

- Normalization first proposed by Codd (1972)
- *Normalization*: The process of decomposing unsatisfactory "bad" relations by breaking up their attributes into smaller relations
- *Normal form*: Condition using keys and FDs of a relation to certify whether a relation schema is in a particular normal form
- Codd proposed 1NF, 2NF, 3NF (normal form)
- A stronger definition of 3NF – Boyce-Codd normal form (BCNF) – was proposed by Boyce and Codd
- 2NF, 3NF, BCNF based on *keys and FDs* of a relation schema

Normalization

- The process of decomposition must also confirm the existence of additional properties:
  - (a) The *lossless join* or *nonadditive join* property – guarantees that the spurious tuples does not occur w.r.t. the relational schemas created after decomposition
  - (b) The *dependency preservation* property – ensures that each FD is preserved in the resulting relations
  - Note that property (a) is extremely important and *cannot* be sacrificed. Property (b) is less stringent and may be sacrificed.

Definition of Keys – Recap

- A **superkey** of a relation schema $R = \{A_1, A_2, ..., A_n\}$ is a set of attributes S *subset-of* R with the property that no two tuples t_1 and t_2 in any legal relation state r of R will have $t_1[S] = t_2[S]$
- A **key** K is a superkey with the *additional property* that removal of any attribute from K will cause K not to be a superkey any more.
- If a relation schema has more than one key, each is called a **candidate key.**
- One of the candidate keys is *arbitrarily* designated to be the **primary key,** and the others are called *secondary keys*.

Definition of Keys – Recap

- A **Prime attribute** must be a member of *some candidate key*
- A **Nonprime attribute** is not a prime attribute—that is, it is not a member of any candidate key.
- Keys are considered from 2NF onwards

First Normal Form – 1NF

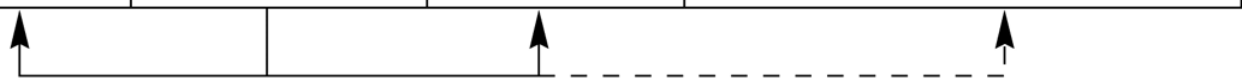
- Domain of an attribute must include only *atomic* (simple, indivisible) values and that the value of any attribute in a tuple must be a *single value*
- Disallows composite attributes, multivalued attributes, and **nested relations**; attributes whose values *for an individual tuple* are non-atomic
- Now considered to be part of the formal definition of a relation

First Normal Form – 1NF

(a)

DEPARTMENT

DNAME	DNUMBER	DMGRSSN	DLOCATIONS



(b)

DEPARTMENT

DNAME	DNUMBER	DMGRSSN	DLOCATIONS
Research	5	333445555	{Bellaire, Sugarland, Houston}
Administration	4	987654321	{Stafford}
Headquarters	1	888665555	{Houston}

(c)

DEPARTMENT

DNAME	DNUMBER	DMGRSSN	DLOCATION
Research	5	333445555	Bellaire
Research	5	333445555	Sugarland
Research	5	333445555	Houston
Administration	4	987654321	Stafford
Headquarters	1	888665555	Houston

First Norma

- Remove DLOCATION
- Place it in a separate relation along with the primary key
- Does not suffer from redundancy

EMPLOYEE

ENAME	SSN	BDATE	ADDRESS	
Smith,John B.	123456789	1965-01-09	731 Fondren,Houston,TX	5
Wong,Franklin T.	333445555	1955-12-08	638 Voss,Houston,TX	5
Zelaya,Alicia J.	999887777	1968-07-19	3321 Castle,Spring,TX	4
Wallace,Jennifer S.	987654321	1941-06-20	291 Berry,Bellaire,TX	4
Narayan,Remesh K.	666884444	1962-09-15	975 Fire Oak,Humble,TX	5
English,Joyce A.	453453453	1972-07-31	5631 Rice,Houston,TX	5
Jabbar,Ahmad V.	987987987	1969-03-29	980 Dallas,Houston,TX	4
Borg,James E.	888665555	1937-11-10	450 Stone,Houston,TX	1

DEPARTMENT

DNAME	DNUMBER	DMGRSSN
Research	5	333445555
Administration	4	987654321
Headquarters	1	888665555

DEPT_LOCATIONS

DNUMBER	DLOCATION
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

SSN	PNUMBER	HOURS
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	null

PROJECT

PNAME	PNUMBER	PLOCATION	DNUM
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

First Normal Form –

- Normalizing nested relations into 1NF
- PROJS is a nested relation – each tuple have a *relation within it*
- Each tuple represents an employee and and a relation PROJS(PNUMBER, HOURS)
- PNUMBER is a partial primary key
- Decompose EMP_PROJ into EMP_PROJ1 and EMP_PROJ2 by *propogating the primary key*

(a)

EMP_PROJ

SSN	ENAME	PROJS	
		PNUMBER	HOURS

(b)

EMP_PROJ

SSN	ENAME	PNUMBER	HOURS
123456789	Smith,John B.	1	32.5
		2	7.5
666884444	Narayan,Ramesh K.	3	40.0
453453453	English,Joyce A.	1	20.0
		2	20.0
333445555	Wong,Franklin T.	2	10.0
		3	10.0
		10	10.0
		20	10.0
999887777	Zelaya,Alicia J.	30	30.0
		10	10.0
987987987	Jabbar,Ahmad V.	10	35.0
		30	5.0
987654321	Wallace,Jennifer S.	30	20.0
		20	15.0
888665555	Borg,James E.	20	null

(c)

EMP_PROJ1

SSN	ENAME

EMP_PROJ2

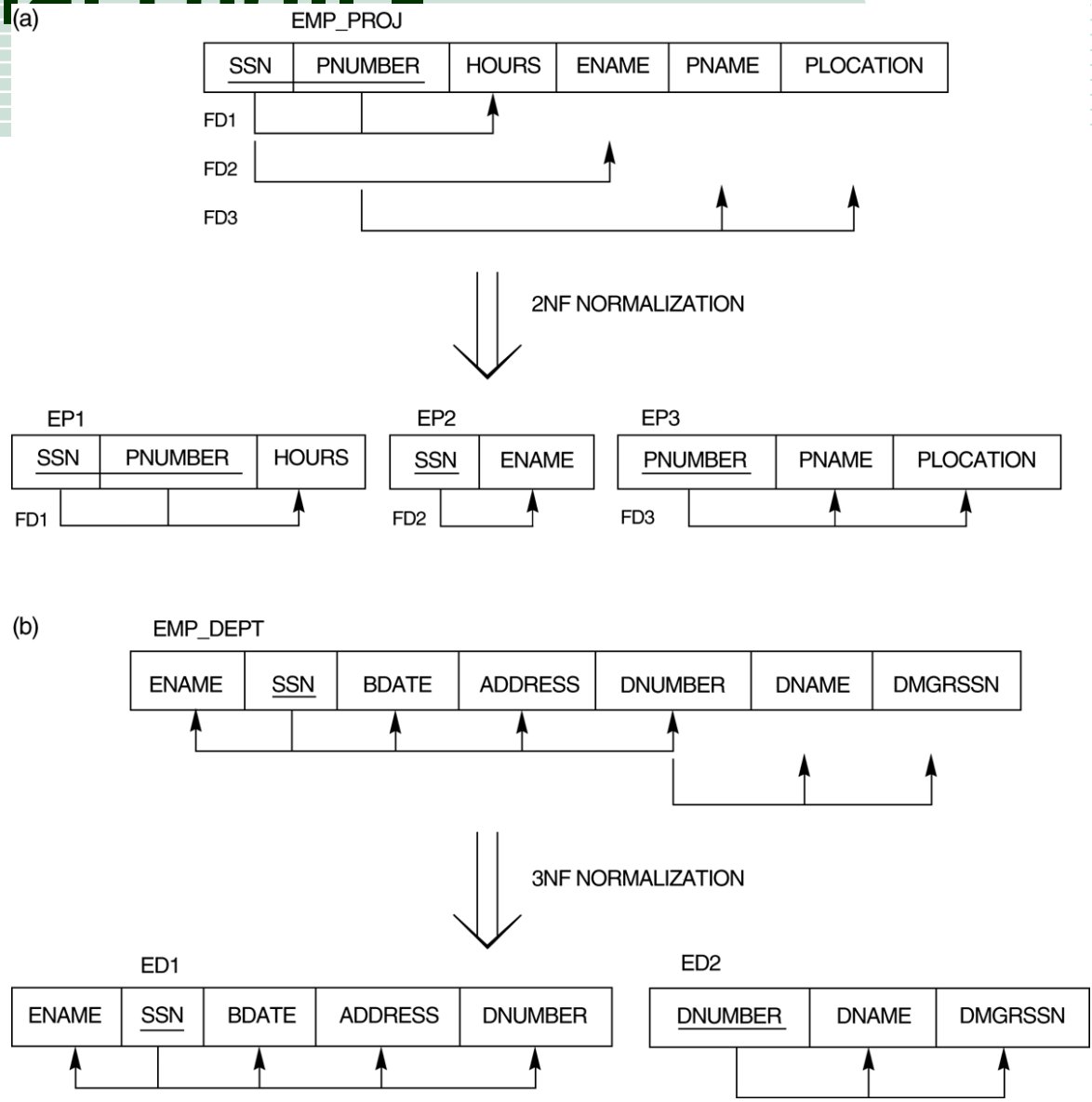
SSN	PNUMBER	HOURS

Second Normal Form – 2NF

- Uses the concepts of **FD**s, **primary key**
- Definitions:
- Prime attribute - attribute that is member of the primary key K
- Full functional dependency – a FD Y -> Z where removal of any attribute from Y means the FD does not hold any more
- A relation schema R is in **second normal form** (**2NF**) if every non-prime attribute A in R is fully functionally dependent on the primary key
- R can be decomposed into 2NF relations via the process of 2NF normalization

Second Normal Form – 2NF

- {SSN, PNUMBER} -> HOURS is a full FD since neither SSN -> HOURS nor PNUMBER -> HOURS hold
- {SSN, PNUMBER} -> ENAME is *not* a full FD (it is called *partial dependency*) since SSN -> ENAME also holds
- FD2 and FD3 violates 2NF



Second Normal Form – 2NF

- Goal: Eliminate reducible FD or Partial FD which violates 2NF.
- Consider R { A,B,C,D } where primary key {A,B}

  /* assume A → D holds which violates 2NF */
- Decompose R into R1 and R2

  R1 { A, D }, primary key { A }

  R2 { A, B, C } primary key { A, B }, foreign key { A } references R1

Second Normal Form – 2NF

- **Update anomalies** in 1NF because of partial FDs.
- {SSN, PUMBER, HOURS, ENAME, PNAME, PLOCATION}
- Difficulty in update operations such as Insert, Delete, and Update.
- Cant insert a new project details without an employee
- Cant delete an employee working lonely in a project, which will delete the corresponding project details also.
- Update on redundant values.

Second Normal Form – 2NF

- Eliminate reducible FD or Partial FD.
- FD2: {SSN} -> ENAME is *not* a full FD (it is called *partial dependency*)
- FD3: {PNUMBER} -> {PNAME, PLOCATION} is *not* a full FD
- FD2 and FD3 violates 2NF.
- R1: {SSN, ENAME}, primary key {SSN}
- R2: {PNUMBER, PNAME, PLOCATION}, primary key {PNUMBER}
- R3: {SSN, PNUMBER, HOURS}, primary key {SSN, PNUMBER}

  SSN references R1(SSN), PNUMBER references R2(PNUMBER)

Third Normal Form – 3NF

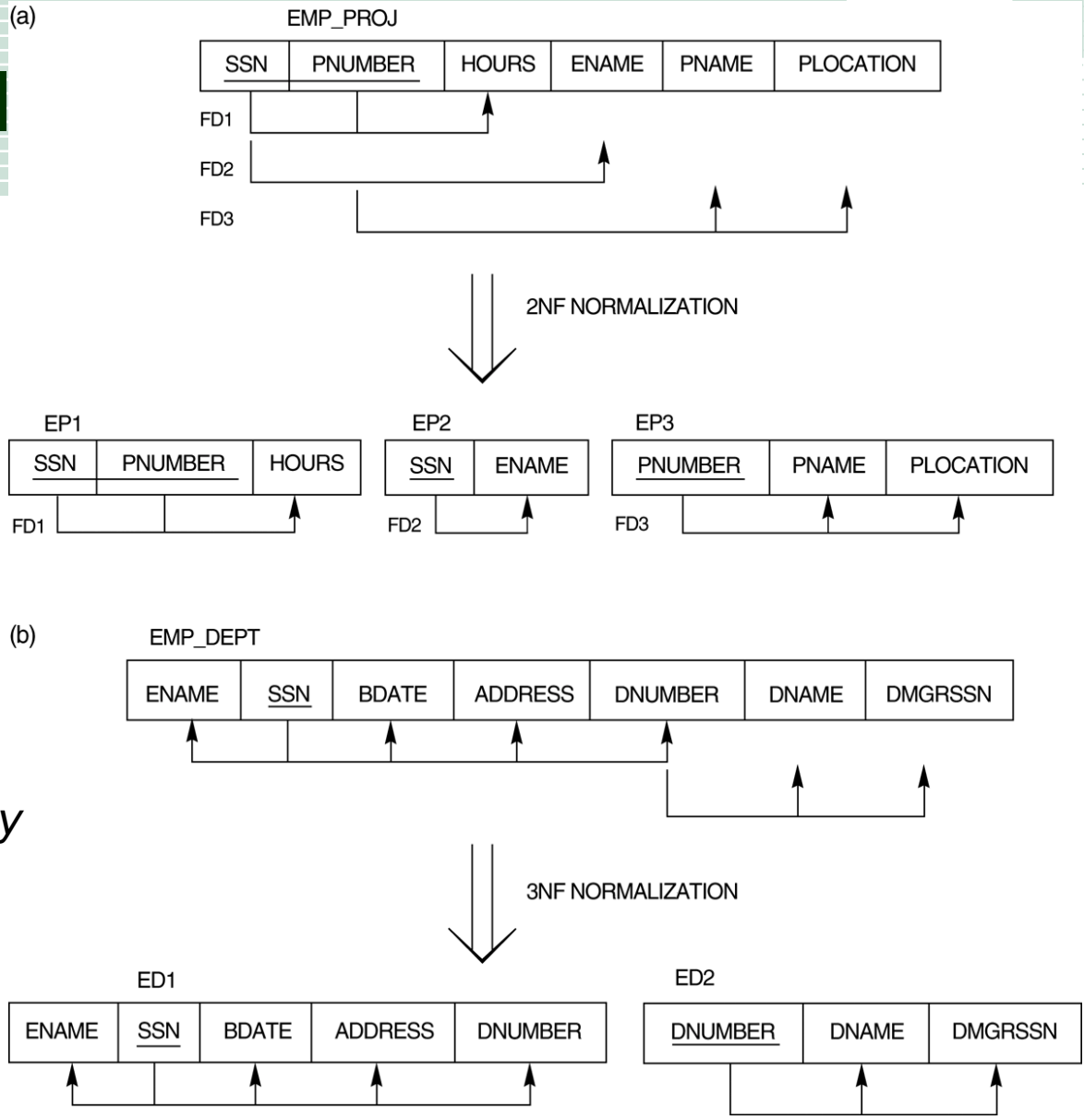
- Uses the concepts of **FD**s, **primary key**
- Definitions:
- **Transitive functional dependency –** a FD X -> Z that can be derived from two FDs X -> Y and Y -> Z
- A FD X -> Z in a R is a *transitive dependency* if there is a set of attributes Y that is neither a candidate key nor a subset of any key of R and both X -> Y and Y -> Z hold
- A relation schema R is in **third normal form** (**3NF**) if it is in 2NF *and* no non-prime attribute A in R is transitively dependent on the primary key

Third Normal Form – 3NF

- Note:
  - In X -> Y and Y -> Z, with X as the primary key, we consider this a problem only if Y is not a candidate key.
  - When Y is a candidate key, there is no problem with the transitive dependency.

Third Normal

- SSN -> DNUMBER and DNUMBER ->DNAME,DMGRSSN hence SSN->DNAME,DMGRSSN is a transitive FD.
- DNUMBER is *neither* a *key nor a subset of the key*
- DNUMBER ->DNAME,DMGRSSN violates 3NF



Third Normal Form – 3NF

- Goal: Eliminate transitive dependencies in 3NF
- Relation not in 3NF suffers from *Update anomalies*:
- Difficulty in update operations such as Insert, Delete, and Update

Third Normal Form – 3NF

- Goal: Eliminate transitive dependencies in 3NF
- R { A,B,C } primary key { A }

  /* assume B → C holds where B is non-prime hence transitive */
- Decompose R into R1 and R2
- R1 { B, C } primary key { B }
- R2 { A, B } primary key { A } foreign key { B } references R1

Third Normal Form – 3NF

- Goal: Eliminate transitive dependencies in 3NF
- R = {SSN, ENAME, BDATE, ADDR, DNUMBER, DNAME, DMGRSSN }

  FD1: SSN → ENAME, BDATE, ADDR, DNUMBER

  FD2: DNUMBER → DNAME, DMGRSSN
- SSN -> DNUMBER and DNUMBER →DNAME,DMGRSSN

  hence SSN→DNAME,DMGRSSN is a transitive FD.
- DNUMBER is *neither* a *key nor a subset of the key*
- FD2 violates 3NF

Third Normal Form – 3NF

- Goal: Eliminate transitive dependencies in 3NF
- R = {SSN, ENAME, BDATE, ADDR, DNUMBER, DNAME, DMGRSSN }

  FD1: SSN → ENAME, BDATE, ADDR, DNUMBER

  FD2: DNUMBER → DNAME, DMGRSSN
- R1 = { SSN, ENAME, BDATE, ADDR, DNUMBER }

  DNUMBER references R2(DNUMBER)
- R2 = { DNUMBER, DNAME, DMGRSSN }

General Normal Form Definitions

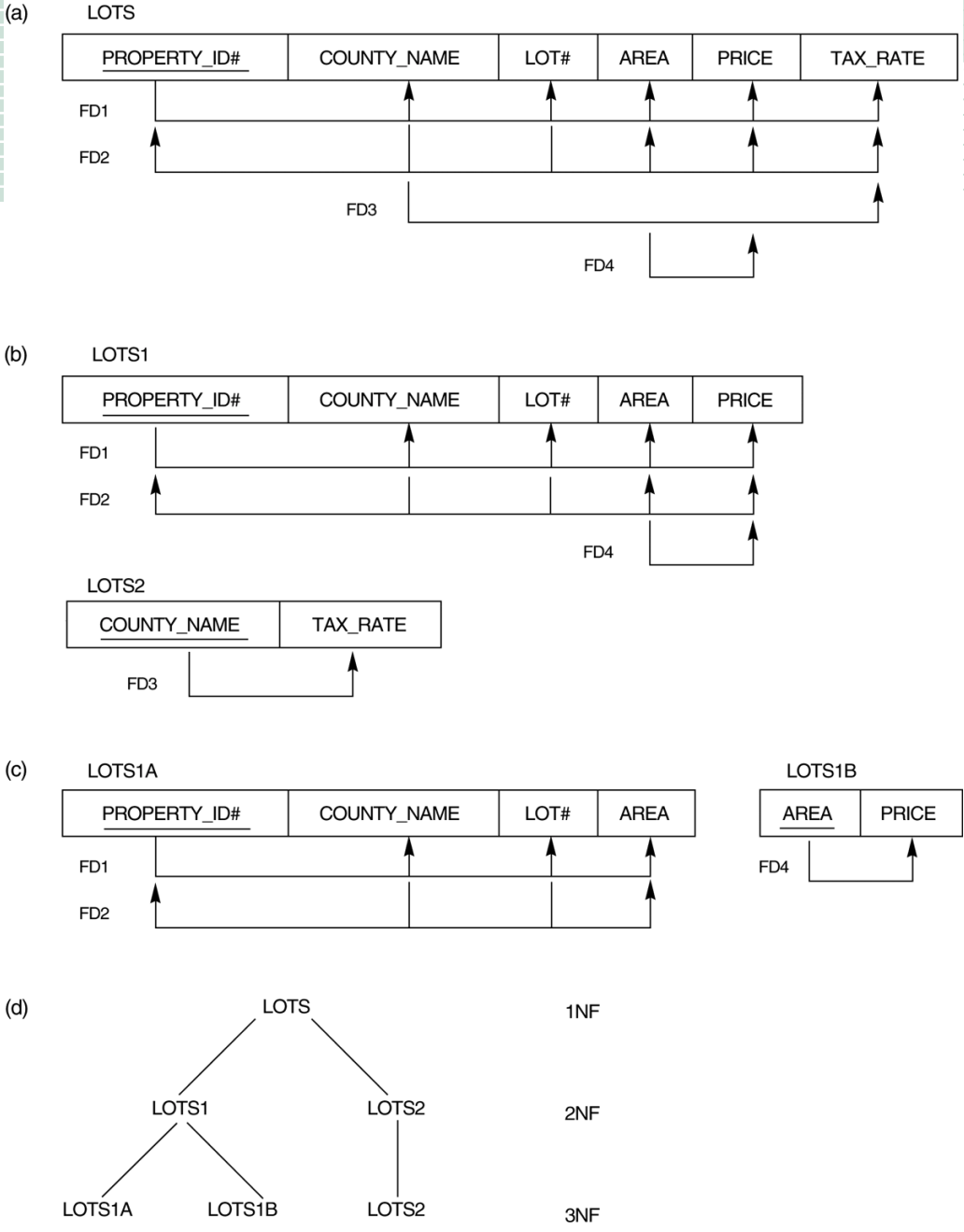
- The above definitions consider the *primary key only*
- The following more general definitions take into account relations with multiple candidate keys
- A relation schema R is in **second normal form** (**2NF**) if every non-prime attribute A in R is fully functionally dependent on *every key* of R.
- A relation schema R is in **third normal form** (**3NF**) if whenever a FD X -> A holds in R, then either:
  - (a) X is a superkey of R, or
  - (b) A is a prime attribute of R

General Normal Form Definitions

- A relation schema R is in **third normal form** (**3NF**) if whenever a FD X -> A holds in R, then either:
  - (a) X is a superkey of R, or
  - (b) A is a prime attribute of R
- Condition (a) checks the two types of dependencies:
- A non-prime attribute determines another non-prime attribute – signals the *transitive FD that violates 3NF*
- A proper subset of a key of R functionally determines a non-prime attribute – signals the *partial FD that violates 2NF*

Normalization

- Two candidate keys: PROPERTY_ID and {COUNTY_NAME, LOT#}
- (a) the LOTS relation with its functional dependencies FD1 though FD4.
- (b) Decomposing into the 2NF relations LOTS1 and LOTS2.
- (c) Decomposing LOTS1 into the 3NF relations LOTS1A and LOTS1B.

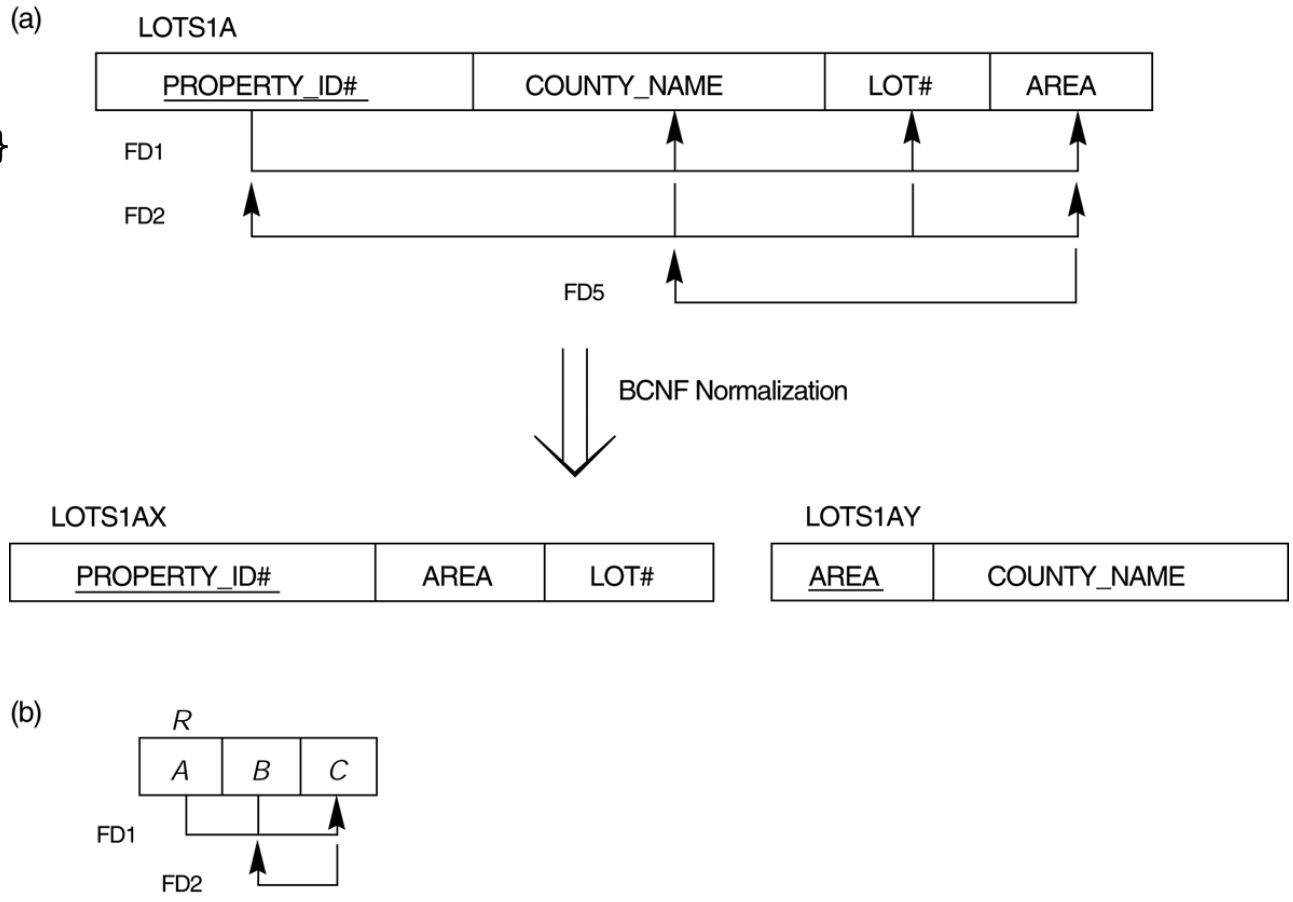


Boyce-Codd Normal Form – BCNF

- A relation schema R is in **Boyce-Codd Normal Form** (**BCNF**) if whenever an FD X -> A holds in R, then X is a superkey of R
- Each normal form is strictly stronger than the previous one
  - Every 2NF relation is in 1NF
  - Every 3NF relation is in 2NF
  - Every BCNF relation is in 3NF
- There exist relations that are in 3NF but not in BCNF
- The goal is to have each relation in BCNF (or 3NF)

Boyce-Codd Normal Form – BCNF

- FD5: {AREA -> COUNTY_NAME}
- FD5 satisfies 3NF but violates BCNF
- AREA is not a superkey of LOTS1A

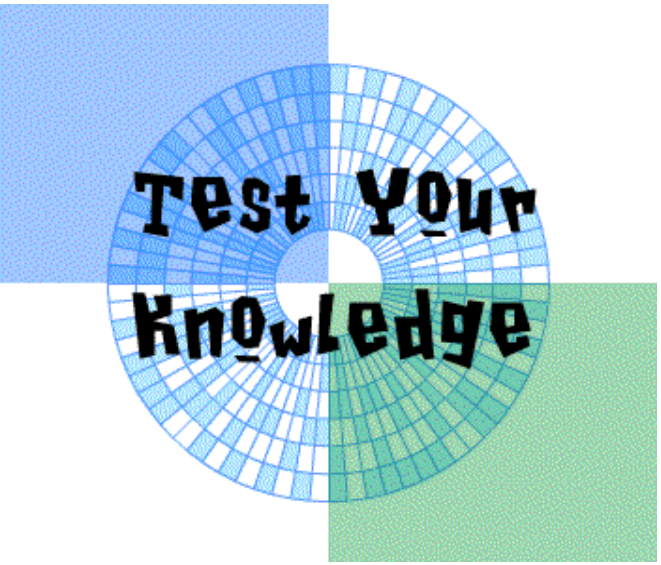


TEACH in 3NF but not in BCNF

- FD1: {STUDENT,COURSE --> INSTRUCTOR}
- FD2: {INSTRUCTOR->COURSE}
- {STUDENT,COURSE} candidate keys
- Decompose TEACH into BCNF

TEACH

STUDENT	COURSE	INSTRUCTOR
Narayan	Database	Mark
Smith	Database	Navathe
Smith	Operating Systems	Ammar
Smith	Theory	Schulman
Wallace	Database	Mark
Wallace	Operating Systems	Ahamad
Wong	Database	Omiecinski
Zelaya	Database	Navathe



References

- Fundamentals of Database Systems, 5th Edition, *Elmasri and Navathe*
- An Introduction to Database Systems, *C.J.Date, Kannan, Swamynathan, 8th Edition*

ACM India 50 Years of Turing Award talk series on

Contributions of Edgar F. Codd by *Prof. Shamkant Navathe, 29th Dec 2017.*

Thank you!