Dr Sophia Tsoka

Department of Informatics

Faculty of Natural and Mathematical Sciences

March 2021

KING'S College LONDON

4CCS1DBS – Database Systems

Week 10 – Functional Dependencies and Normalisation 2

Topic: General Normalisation Aspects, Keys and Superkeys

List of Topics

Term week	Topic	Lecturer
22	Introduction to Database Systems	Dr Tsoka
23	Data Modeling Using the Entity-Relationship Model	Dr Tsoka
24	Relational Data Model	Dr Curcin
25	SQL 1	Dr Curcin
26	SQL 2	Dr Curcin
27	**----CATCH UP WEEK---**	
28	Relational Algebra 1	Dr Curcin
29	Relational Algebra 2	Dr Curcin
30	Normalisation 1	Dr Tsoka
31	Normalisation 2	Dr Tsoka
32	Advanced Topics and Security Issues	Dr Tsoka

This week: Functional Dependencies and Normalisation

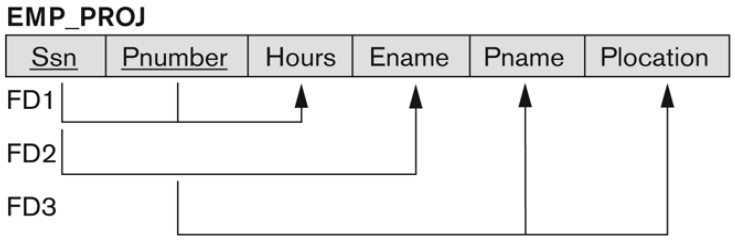
- Normalisation of Relations
- Practical Use of Normal Forms
- Definitions of Keys and Attributes Participating in Keys
- First Normal Form
- Second Normal Form
- Third Normal Form
- BCNF (Boyce-Codd Normal Form)

This recording

- Normalisation of Relations
- Practical Use of Normal Forms
- Definitions of Keys and Attributes Participating in Keys
- First Normal Form
- Second Normal Form
- Third Normal Form
- BCNF (Boyce-Codd Normal Form)

Functional Dependencies - Definition

- Functional dependency (FD) between attributes X and Y of relation R:
  - **X → Y, if whenever two tuples have the same value for X, they *must have* the same value for Y**
  - **for any two tuples t1 and t2 in any relation instance r(R): if t1[X]=t2[X], *then* t1[Y]=t2[Y]**



EMP_PROJ

Ssn	Pnumber	Hours	Ename	Pname	Plocation
123456789	1	32.5	Smith, John B.	ProductX	Bellaire
123456789	2	7.5	Smith, John B.	ProductY	Sugarland
666884444	3	40.0	Narayan, Ramesh K.	ProductZ	Houston
453453453	1	20.0	English, Joyce A.	ProductX	Bellaire
453453453	2	20.0	English, Joyce A.	ProductY	Sugarland
333445555	2	10.0	Wong, Franklin T.	ProductY	Sugarland
333445555	3	10.0	Wong, Franklin T.	ProductZ	Houston
333445555	10	10.0	Wong, Franklin T.	Computerization	Stafford
333445555	20	10.0	Wong, Franklin T.	Reorganization	Houston
999887777	30	30.0	Zelaya, Alicia J.	Newbenefits	Stafford
999887777	10	10.0	Zelaya, Alicia J.	Computerization	Stafford
987987987	10	35.0	Jabbar, Ahmad V.	Computerization	Stafford
987987987	30	5.0	Jabbar, Ahmad V.	Newbenefits	Stafford
987654321	30	20.0	Wallace, Jennifer S.	Newbenefits	Stafford
987654321	20	15.0	Wallace, Jennifer S.	Reorganization	Houston
888665555	20	Null	Borg, James E.	Reorganization	Houston

Definition of Keys and Superkeys of a Relation

- **Superkey** of R:
  - Is a set of attributes SK of R with the following condition:
    - No two tuples in any valid relation state r(R) will have the same value for SK
    - That is, for any distinct tuples t1 and t2 in r(R), t1[SK] ≠ t2[SK]
    - This condition must hold in *any valid state* r(R)
- **Key** of R:
  - A "minimal" superkey
  - That is, a key is a superkey K such that removal of any attribute from K results in a set of attributes that is not a superkey (does not possess the superkey uniqueness property)

EMP_PROJ

Ssn	Pnumber	Hours	Ename	Pname	Plocation
123456789	1	32.5	Smith, John B.	ProductX	Bellaire
123456789	2	7.5	Smith, John B.	ProductY	Sugarland
666884444	3	40.0	Narayan, Ramesh K.	ProductZ	Houston
453453453	1	20.0	English, Joyce A.	ProductX	Bellaire
453453453	2	20.0	English, Joyce A.	ProductY	Sugarland
333445555	2	10.0	Wong, Franklin T.	ProductY	Sugarland
333445555	3	10.0	Wong, Franklin T.	ProductZ	Houston
333445555	10	10.0	Wong, Franklin T.	Computerization	Stafford
333445555	20	10.0	Wong, Franklin T.	Reorganization	Houston
999887777	30	30.0	Zelaya, Alicia J.	Newbenefits	Stafford
999887777	10	10.0	Zelaya, Alicia J.	Computerization	Stafford
987987987	10	35.0	Jabbar, Ahmad V.	Computerization	Stafford
987987987	30	5.0	Jabbar, Ahmad V.	Newbenefits	Stafford
987654321	30	20.0	Wallace, Jennifer S.	Newbenefits	Stafford
987654321	20	15.0	Wallace, Jennifer S.	Reorganization	Houston
888665555	20	Null	Borg, James E.	Reorganization	Houston

Keys and Superkeys of Relation CAR

- Example: Consider the CAR relation schema:
  - CAR(State, Reg#, SerialNo, Make, Model, Year)
  - Define superkeys of relation CAR
  - Define keys of relation CAR

CAR_1
((ABC 123, TEXAS), TK629, Ford Mustang, convertible, 2004 {red, black})

CAR_2
((ABC 123, NEW YORK), WP9872, Nissan Maxima, 4-door, 2005, {blue})

CAR_3
((VSY 720, TEXAS), TD729, Chrysler LeBaron, 4-door, 2002, {white, blue})

⋮

Keys and Superkeys of Relation CAR

- Example: Consider the CAR relation schema:
  - CAR(State, Reg#, SerialNo, Make, Model, Year)
  - CAR has two keys:
    - Key1 = {State, Reg#}
    - Key2 = {SerialNo}
  - Both are also superkeys of CAR
  - {SerialNo, Make} is a superkey but *not* a key.
- In general:
  - Any *key* is a *superkey* (but not vice versa)
  - Any set of attributes that *includes a key* is a *superkey*
  - A *minimal* superkey is also a key

CAR_1
((ABC 123, TEXAS), TK629, Ford Mustang, convertible, 2004 {red, black})

CAR_2
((ABC 123, NEW YORK), WP9872, Nissan Maxima, 4-door, 2005, {blue})

CAR_3
((VSY 720, TEXAS), TD729, Chrysler LeBaron, 4-door, 2002, {white, blue})

⋮

Definitions of Keys and Attributes Participating in Keys

- 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**.

- ***Prime*** is an attribute that is member of *some* candidate key

- ***Nonprime*** is an attribute that is not a prime attribute - i.e. it is not a member of *any* candidate key.

Prime and Nonprime attributes

- Using the previous example of the CAR relation, as stated previously:
  - CAR(State, Reg#, SerialNo, Make, Model, Year)
  - CAR has two keys:
    - Key1 = {State, Reg#}
    - Key2 = {SerialNo}
- Therefore:
  - Prime attributes are: SerialNo, State, Reg#
  - Nonprime attributes are: Make, Model, Year

Normalisation of Relations

- **Normalisation definition**
  - The process of:
    - analysing relation schemas based on FDs and candidate keys to minimise redundancy and insertion, deletion and update anomalies
    - decomposing unsatisfactory 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

Normalisation of Relations

- 2NF, 3NF, BCNF
  - based on keys and FDs of a relation schema

- 4NF based on keys, multi-valued dependencies (MVDs); 5NF based on keys, join dependencies (JDs) – not covered here

Practical Use of Normal Forms

- **Normalisation** is carried out in practice so that the resulting designs are of high quality and meet the desirable properties
- The practical utility of these normal forms becomes questionable when the constraints on which they are based are *hard to understand* or to *detect*
- The database designers *need not* normalize to the highest possible normal form
  - usually up to 3NF or BCNF

Dr Sophia Tsoka

Department of Informatics

Faculty of Natural and Mathematical Sciences

March 2021

KING'S College LONDON

4CCS1DBS – Database Systems

Week 10 – Functional Dependencies and Normalisation 2

Topic: First, Second and Third Normal Form

This week: Functional Dependencies and Normalisation

- Normalisation of Relations
- Practical Use of Normal Forms
- Definitions of Keys and Attributes Participating in Keys
- First Normal Form
- Second Normal Form
- Third Normal Form
- BCNF (Boyce-Codd Normal Form)

First Normal Form

- Disallows
  - composite attributes
  - multivalued attributes
  - **nested relations**; attributes whose values for an *individual tuple* are non-atomic

- Considered to be part of the definition of relation.

Normalisation into 1NF

(a)

DEPARTMENT

Dname	Dnumber	Dmgr_ssn	Dlocations

(b)

DEPARTMENT

Dname	Dnumber	Dmgr_ssn	Dlocations
Research	5	333445555	{Bellaire, Sugarland, Houston}
Administration	4	987654321	{Stafford}
Headquarters	1	888665555	{Houston}

(c)

DEPARTMENT

Dname	Dnumber	Dmgr_ssn	Dlocation
Research	5	333445555	Bellaire
Research	5	333445555	Sugarland
Research	5	333445555	Houston
Administration	4	987654321	Stafford
Headquarters	1	888665555	Houston

To normalise: remove attribute that causes the problem and place in separate relation together with the primary key

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

Normalisation of Nested Relations into 1NF

(a)

EMP_PROJ

		Projs	
Ssn	Ename	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, AliciaJ.	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

Nested relations, as with composite attributes, are disallowed under 1NF.

Relation EMP_PROJ is NOT in 1NF.

Primary keys: SSN and Pnumber within nested relation.

To normalise: remove nested relation and place in separate relation together with the primary key, as in (c).

Second Normal Form – Necessary Definitions

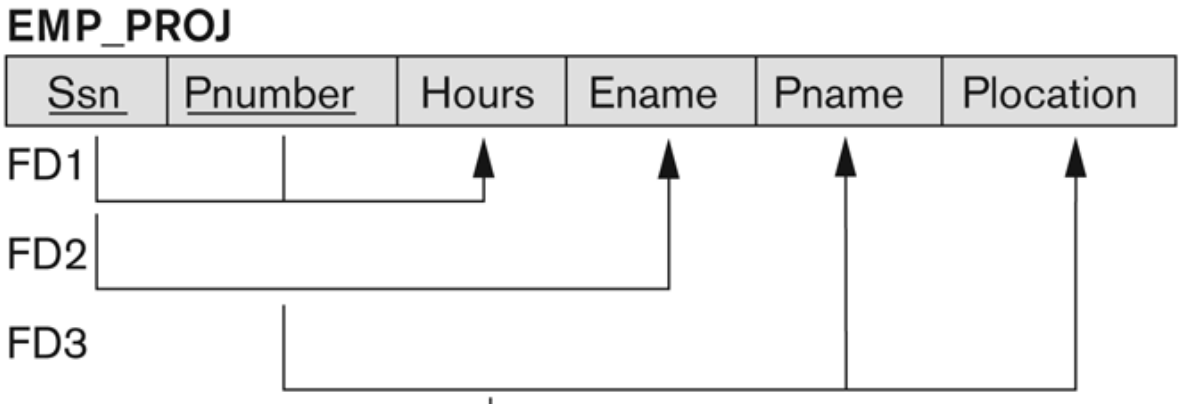
- **Prime attribute:** An attribute that is member of any candidate key

- **Full functional dependency:** a FD Y→ Z where removal of any attribute from Y means the FD does not hold any more
- Examples: {SSN, PNUMBER} → HOURS
  - Full FD
  - since neither SSN → HOURS nor PNUMBER → HOURS hold
- {SSN, PNUMBER} → ENAME
  - It is not a full FD
  - since SSN → ENAME also holds
  - it is called a *partial* dependency

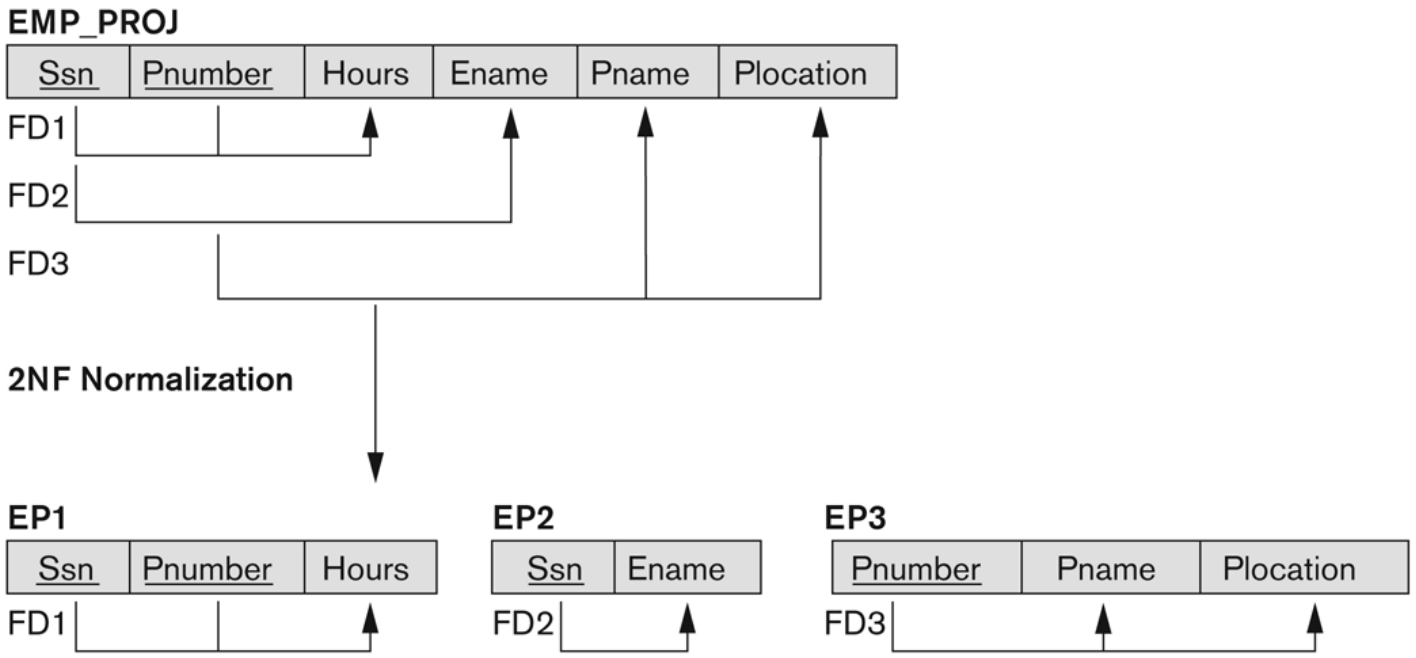
Second Normal Form

- 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
- Test that left-hand side in FD is part of primary key.
- If the primary key contains a single attribute, the test need not be applied at all.
- Relation, R, can be decomposed into 2NF relations via the process of 2NF normalisation

Is this relation in 2NF?



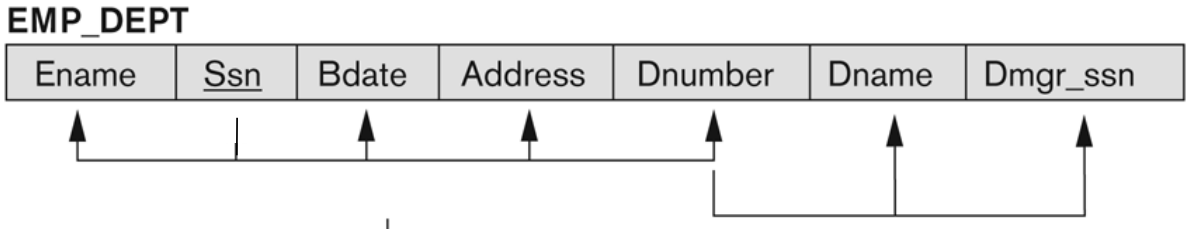
Normalising into 2NF



Third Normal Form - Necessary Definition

- **Definition:**
  - **Transitive functional dependency:**
    - a FD X → Y that can be derived from two FDs: X → Z and Z → Y

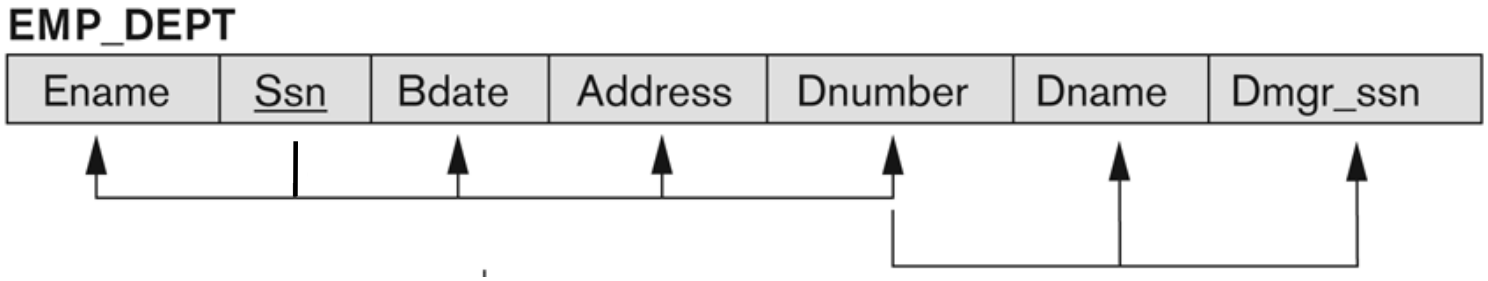
- **Examples:**
  - SSN → DMGRSSN is a **transitive** FD
    - SSN → DNUMBER and DNUMBER → DMGRSSN hold
  - SSN → ENAME is **non-transitive**
    - Since there is no set of attributes X where SSN → X and X→ ENAME



Third Normal Form

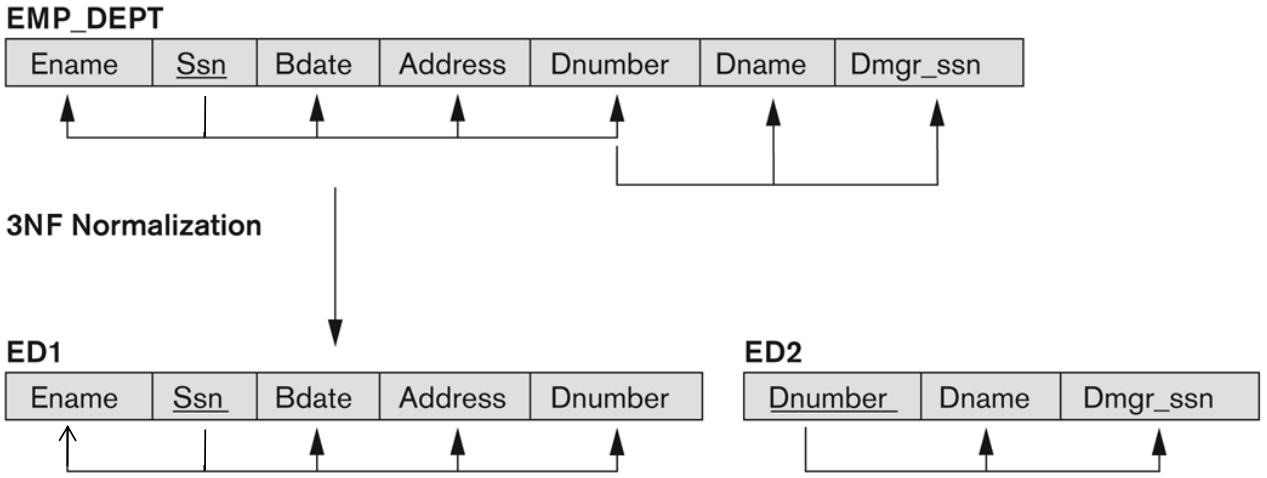
- A relation schema R is in **third normal form (3NF)** if
  - it is in 2NF *and*
  - no non-prime attribute A in R is transitively dependent on the primary key
- R can be decomposed into 3NF relations via the process of 3NF normalisation
- NOTE:
  - In X → Y and Y → Z, with X as the primary key, we 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.

Is this relation in 3NF ?



- 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

Normalisation into 3NF



Dr Sophia Tsoka

Department of Informatics

Faculty of Natural and Mathematical Sciences

March 2021

KING'S College LONDON

4CCS1DBS – Database Systems

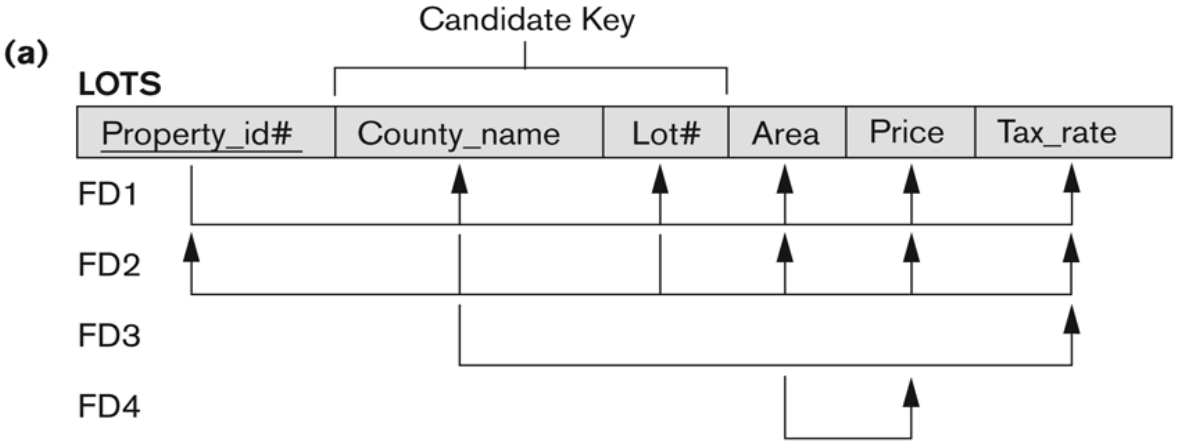
Week 10 – Functional Dependencies and Normalisation 2

Topic: Normalisation Process from 1NF to 3NF and Boyce-Codd Normal Form

This recording

- Normalisation of Relations
- Practical Use of Normal Forms
- Definitions of Keys and Attributes Participating in Keys
- First Normal Form
- Second Normal Form
- Third Normal Form
- BCNF (Boyce-Codd Normal Form)

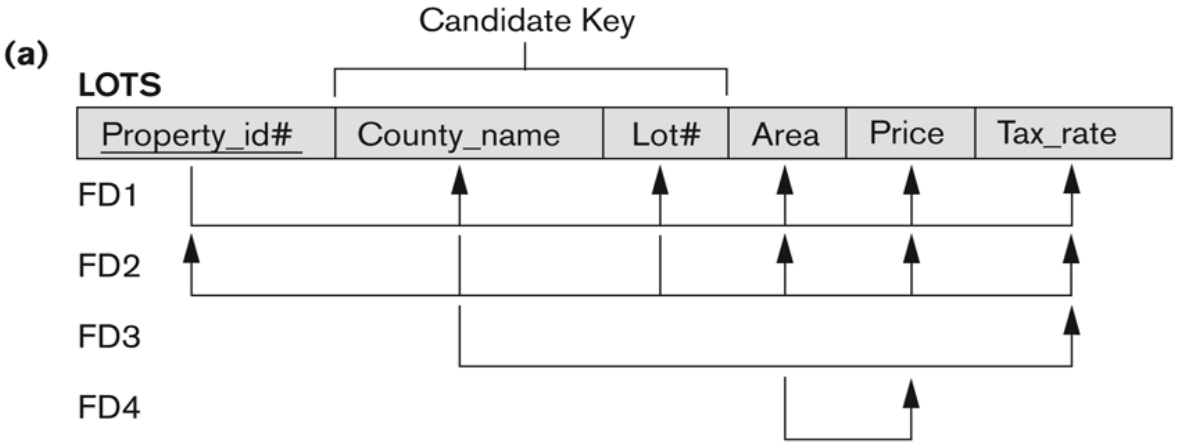
The LOTS Relation



- The LOTS relation describes parcels of land for sale in various counties

- Candidate Key 1: {Property_id#} (we choose this to be the primary key)
- Candidate Key 2: {County_name, Lot#}, i.e. lot numbers are unique ONLY within each county

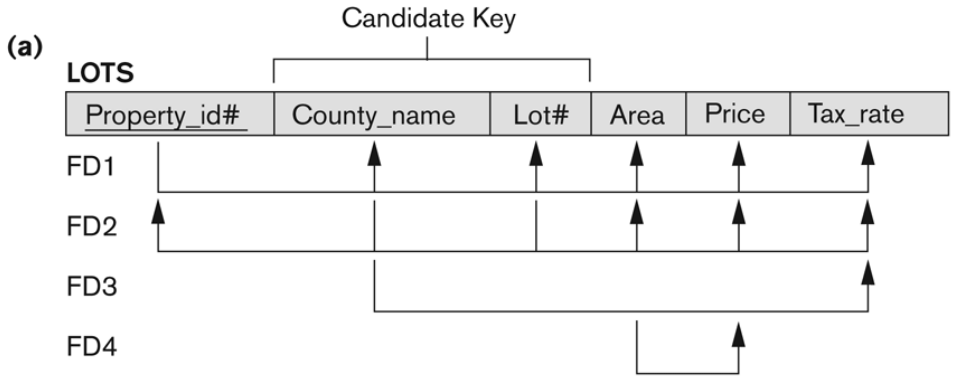
The LOTS Relation



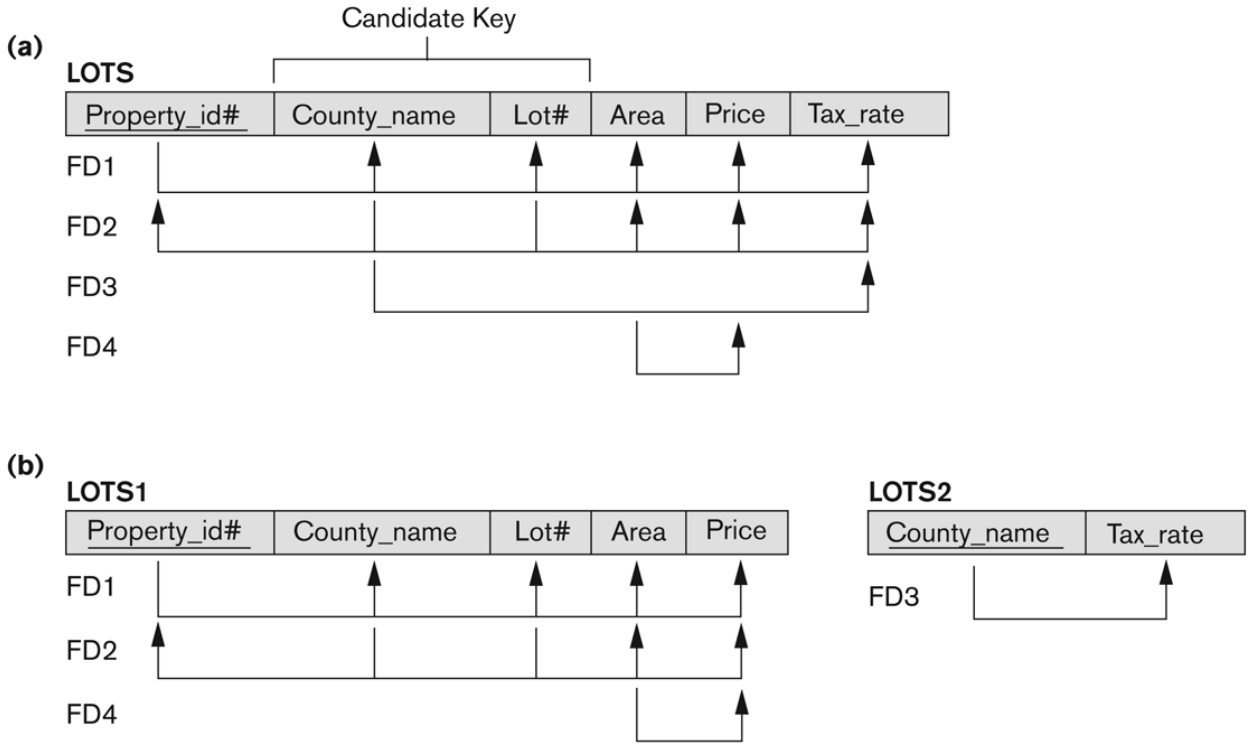
- FD3: tax rate is fixed for given county
- FD4: the price of a lot is determined by its area (regardless of which county it is in).

Normalisation of LOTS into 2NF

- A relation schema R is in **second normal form (2NF)** if every nonprime attribute A in R is not *partially* dependent on any key of R.
  - Test for FDs whose left-hand side attributes are ***part*** of the primary key.

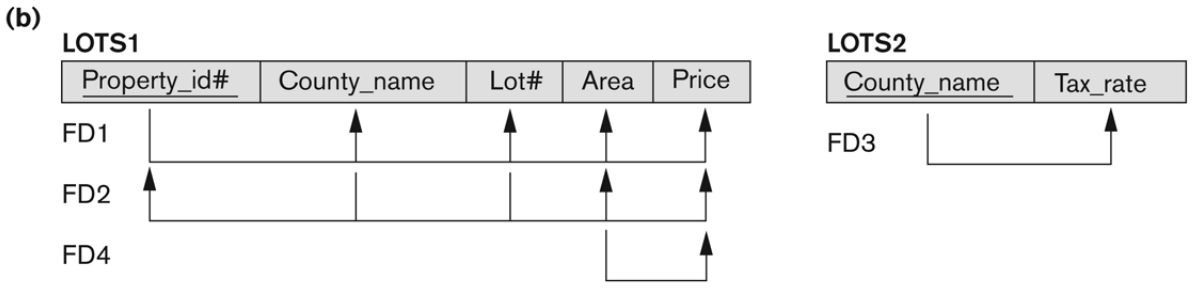


Normalisation of LOTS into 2NF

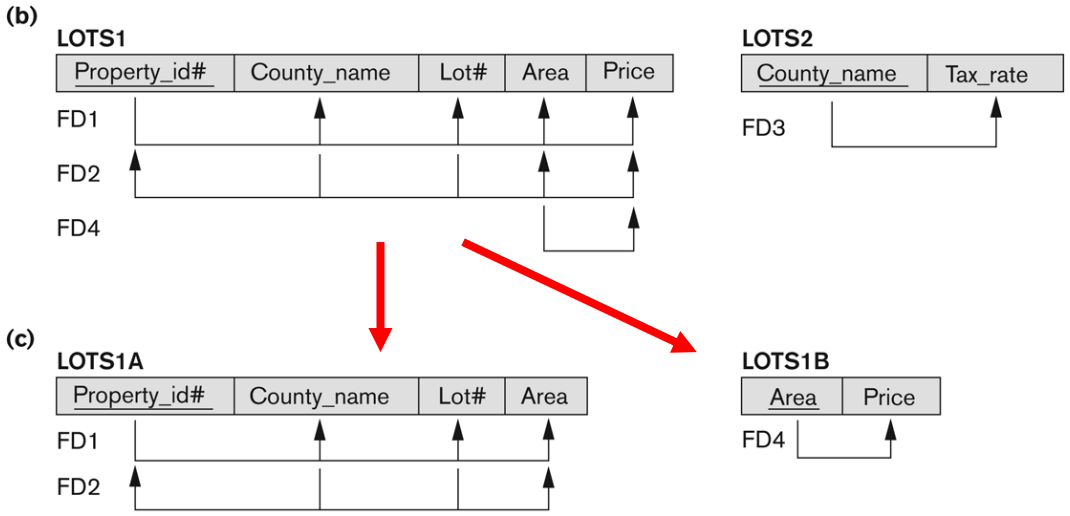


Normalisation of LOTS1 into 3NF

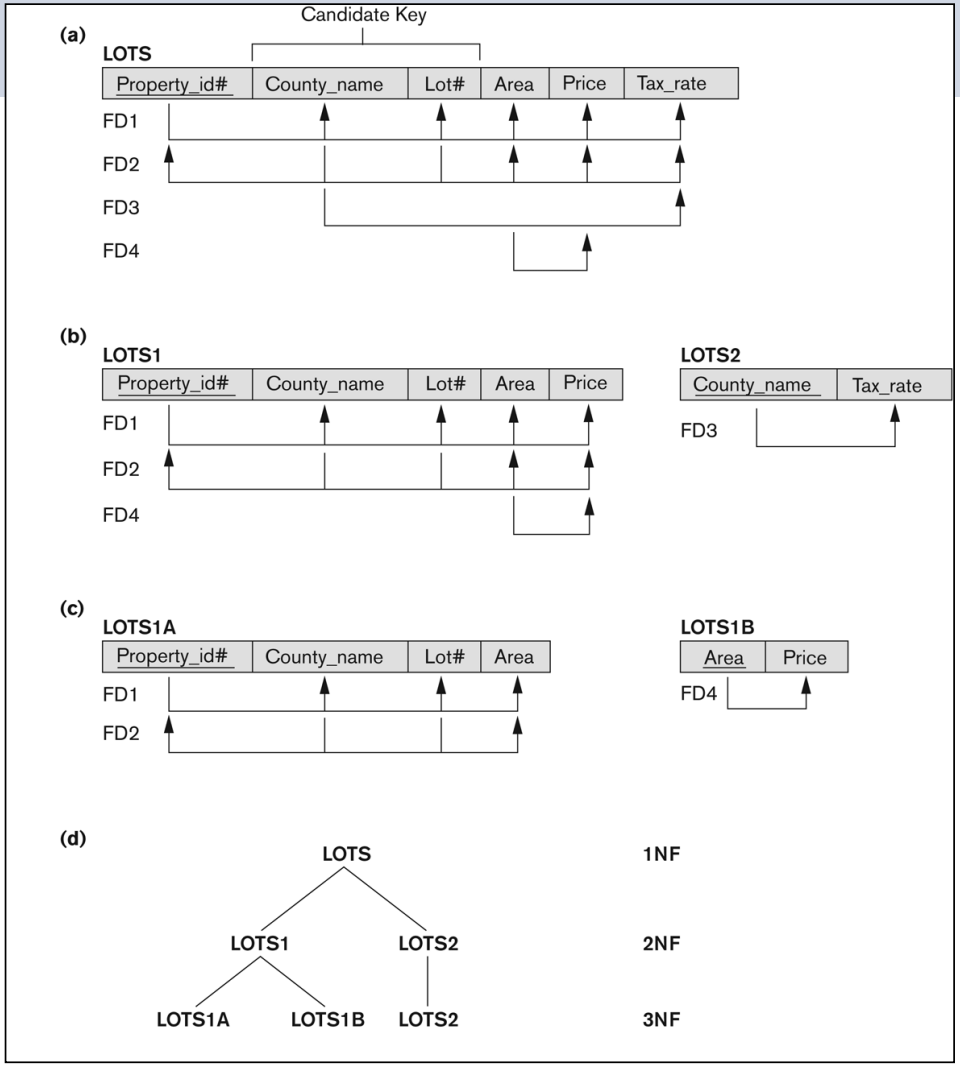
- A relation schema R is in **third normal form (3NF)** if, whenever a nontrivial FD X → A holds in R, either
- (a) X is a superkey of R, or
- (b) A is a prime attribute of R.



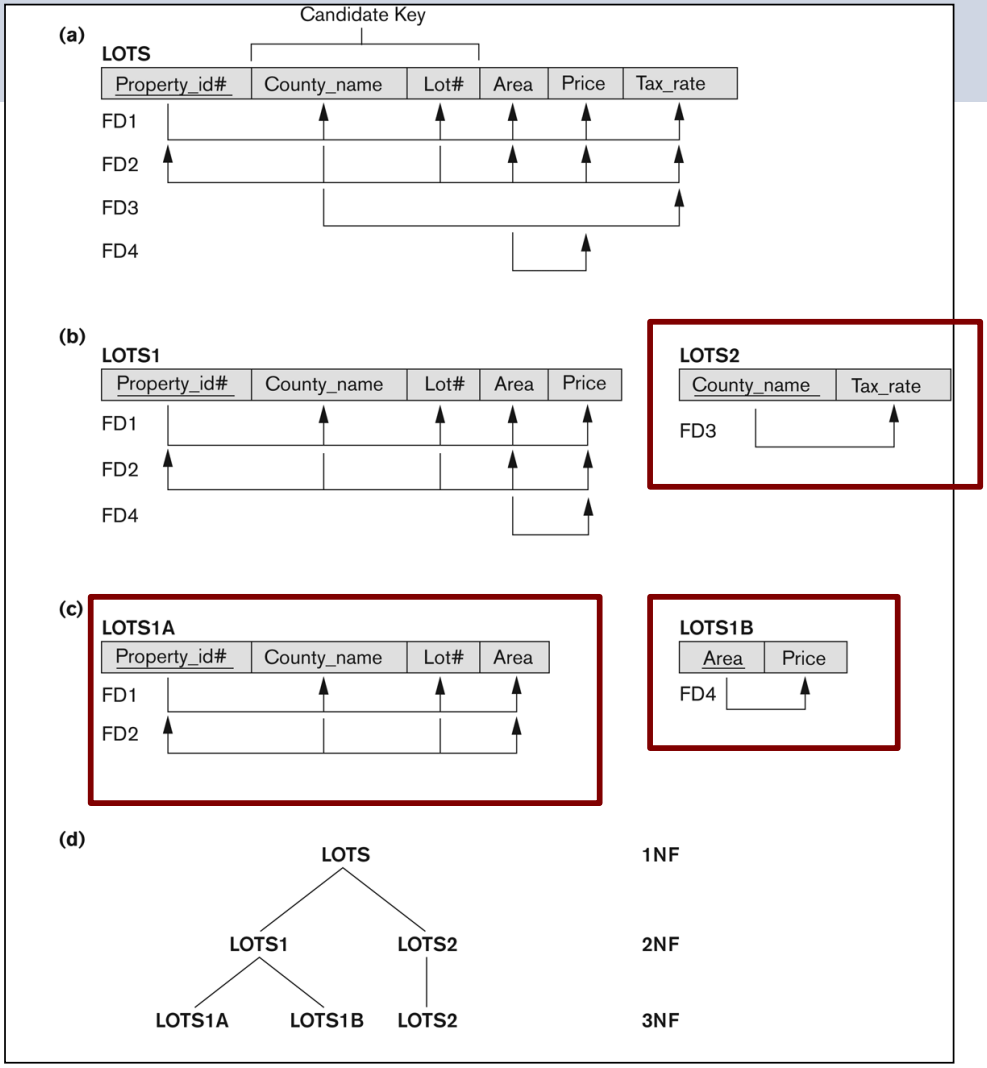
Normalisation of LOTS into 3NF



Successive Normalisation of LOTS into 2NF and 3NF



Successive Normalisation of LOTS into 2NF and 3NF



SUMMARY OF NORMAL FORMS based on Primary Keys

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).

Alternative 3NF Definition

- **Superkey** of relation schema R - a set of attributes S of R that contains a 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

- NOTE: Boyce-Codd normal form disallows condition (b) above

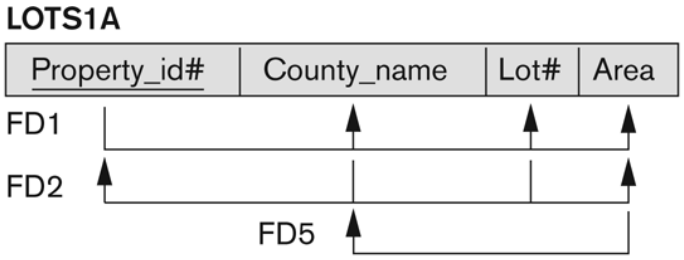
BCNF (Boyce-Codd Normal Form)

- 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

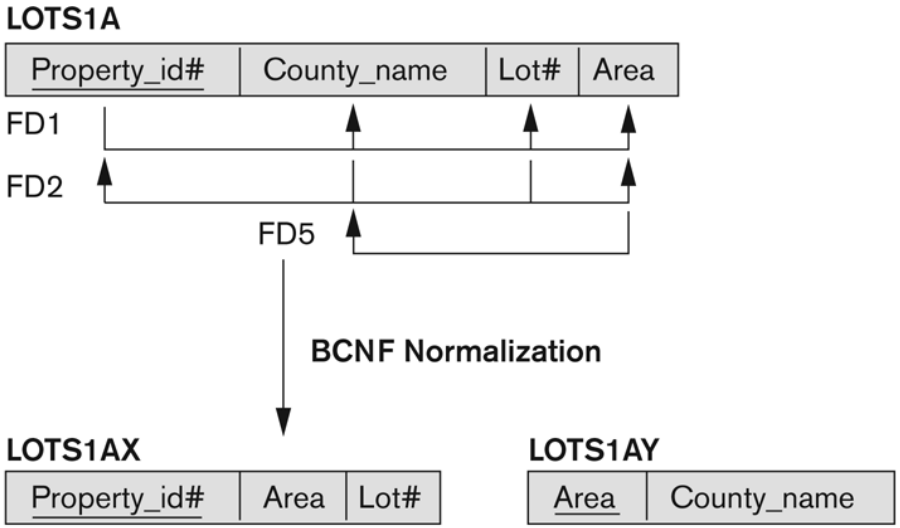
- There exist relations that are in 3NF but not in BCNF
- The goal is to have each relation in BCNF (or 3NF)

Boyce-Codd Normal Form Example



- Suppose that lots in particular counties are of a specific area.
- Then FD Area → County_name holds.
- Is the relation in 3NF? In the relation in BCNF?

Boyce-Codd Normal Form Example



- FD Area → County_name holds.
- The relation is in 3NF but not in BCNF, so decomposition as shown is required.

Boyce-Codd Normal Form

- In practice, most relation schemata that are in 3NF are also in BCNF.
- Only if X → A holds in relation schema R with X *not* being a superkey *and* A being a prime attribute, then R will be in 3NF but NOT in BCNF.

Summary

- Normalisation of Relations
- Practical Use of Normal Forms
- Definitions of Keys and Attributes Participating in Keys
- First Normal Form
- Second Normal Form
- Third Normal Form
- BCNF (Boyce-Codd Normal Form)