CS4.301 Data & Applications

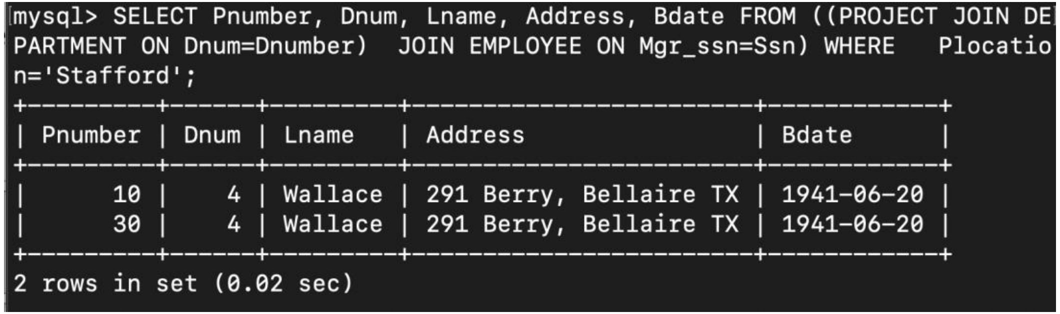
Ponnurangam Kumaraguru ("PK")
#ProfGiri @ IIIT Hyderabad

pk.profgiri /in/ponguru @ponguru Ponnurangam.kumaraguru

Multiway JOIN in the FROM clause

Can nest JOIN specifications for a multiway join:

SELECT Pnumber, Dnum, Lname, Address, Bdate FROM ((PROJECT JOIN DEPARTMENT ON Dnum=Dnumber) JOIN EMPLOYEE ON Mgr_ssn=Ssn) WHERE Plocation='Stafford';



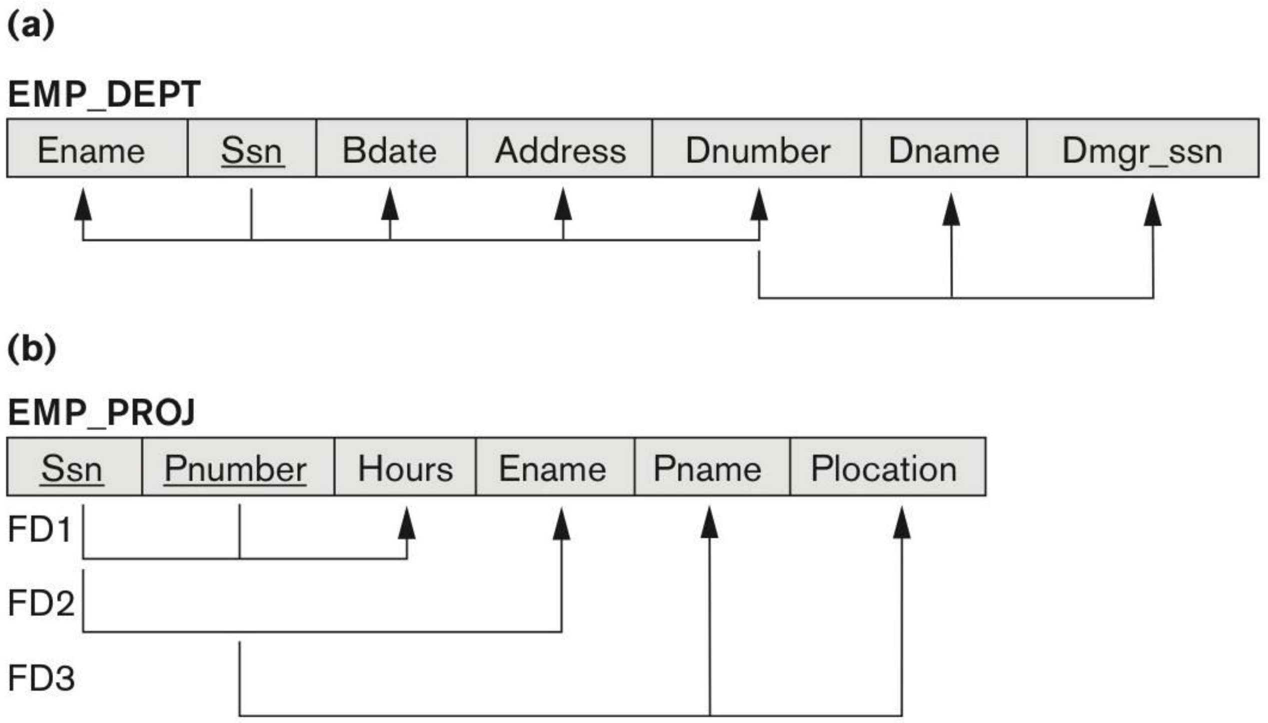
```
mysql> SELECT Pnumber, Dnum, Lname, Address, Bdate FROM ((PROJECT JOIN DE
PARTMENT ON Dnum=Dnumber)  JOIN EMPLOYEE ON Mgr_ssn=Ssn) WHERE   Plocatio
n='Stafford';
+---------+------+---------+-------------------------+------------+
| Pnumber | Dnum | Lname   | Address                 | Bdate      |
+---------+------+---------+-------------------------+------------+
|      10 |    4 | Wallace | 291 Berry, Bellaire TX | 1941-06-20 |
|      30 |    4 | Wallace | 291 Berry, Bellaire TX | 1941-06-20 |
+---------+------+---------+-------------------------+------------+
2 rows in set (0.02 sec)
```

1.1 Semantics of the Relational Attributes must be clear

GUIDELINE 1: Informally, each tuple in a relation should represent one entity or relationship instance. (Applies to individual relations and their attributes).

- Attributes of different entities (EMPLOYEEs, DEPARTMENTs, PROJECTs) should not be mixed in the same relation
- Only foreign keys should be used to refer to other entities

Bottom Line: *Design a schema that can be explained easily relation by relation. The semantics of attributes should be easy to interpret.*



(a)

EMP_DEPT

Ename	Ssn	Bdate	Address	Dnumber	Dname	Dmgr_ssn

(b)

EMP_PROJ

Ssn	Pnumber	Hours	Ename	Pname	Plocation

Any concerns here?

EMP_DEPT: mixing attributes of employees & departments

EMP_PROJ: mixes attributes of employees, projects & works_on

1.2 Redundant Information in Tuples and Update Anomalies

- Information is stored redundantly
  - Wastes storage
  - Causes problems with update anomalies
    - Insertion anomalies
    - Deletion anomalies
    - Modification anomalies

Guideline for Redundant Information in Tuples and Update Anomalies

GUIDELINE 2:

Design a schema that does not suffer from the insertion, deletion and update anomalies

If there are any anomalies present, then note them so that applications can be made to take them into account

1.3 Null Values in Tuples

GUIDELINE 3:

- Relations should be designed such that their tuples will have as few NULL values as possible
- Attributes that are NULL frequently could be placed in separate relations (with the primary key)

Reasons for nulls; different meanings for null:

- Attribute not applicable or invalid [visa status to US students]
- Attribute value unknown [DOB of an employee]
- Value is known but absent; it has not been recorded yet [phone # of employee]

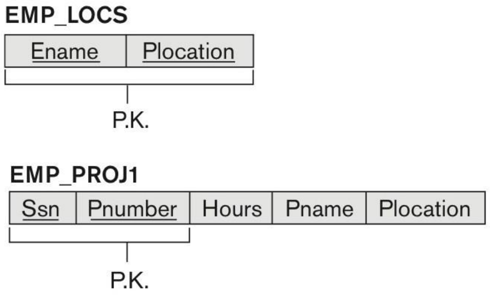
1.4 Generation of Spurious Tuples – avoid at any cost

Bad designs for a relational database may result in erroneous results for certain JOIN operations

GUIDELINE 4:

> No spurious tuples should be generated by doing a natural-join of any relations.

(a)



EMP_LOCS

Ename	Plocation

P.K.

EMP_PROJ1

Ssn	Pnumber	Hours	Pname	Plocation

P.K.

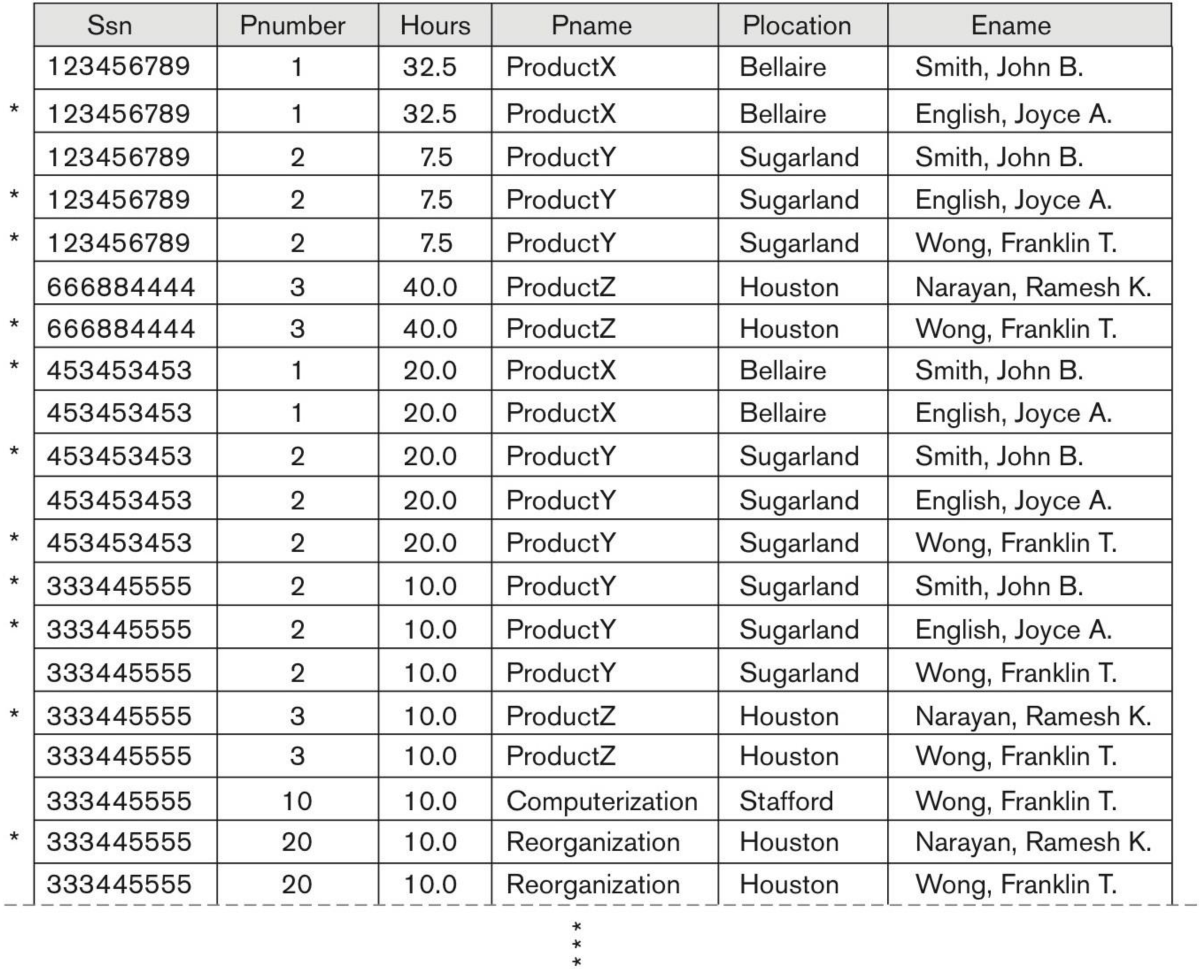
(b)

EMP_LOCS

Ename	Plocation
Smith, John B.	Bellaire
Smith, John B.	Sugarland
Narayan, Ramesh K.	Houston
English, Joyce A.	Bellaire
English, Joyce A.	Sugarland
Wong, Franklin T.	Sugarland
Wong, Franklin T.	Houston
Wong, Franklin T.	Stafford
Zelaya, Alicia J.	Stafford
Jabbar, Ahmad V.	Stafford
Wallace, Jennifer S.	Stafford
Wallace, Jennifer S.	Houston
Borg, James E.	Houston

EMP_PROJ1

Ssn	Pnumber	Hours	Pname	Plocation
123456789	1	32.5	ProductX	Bellaire
123456789	2	7.5	ProductY	Sugarland
666884444	3	40.0	ProductZ	Houston
453453453	1	20.0	ProductX	Bellaire
453453453	2	20.0	ProductY	Sugarland
333445555	2	10.0	ProductY	Sugarland
333445555	3	10.0	ProductZ	Houston
333445555	10	10.0	Computerization	Stafford
333445555	20	10.0	Reorganization	Houston
999887777	30	30.0	Newbenefits	Stafford
999887777	10	10.0	Computerization	Stafford
987987987	10	35.0	Computerization	Stafford
987987987	30	5.0	Newbenefits	Stafford
987654321	30	20.0	Newbenefits	Stafford
987654321	20	15.0	Reorganization	Houston
888665555	20	NULL	Reorganization	Houston



	Ssn	Pnumber	Hours	Pname	Plocation	Ename
	123456789	1	32.5	ProductX	Bellaire	Smith, John B.
*	123456789	1	32.5	ProductX	Bellaire	English, Joyce A.
	123456789	2	7.5	ProductY	Sugarland	Smith, John B.
*	123456789	2	7.5	ProductY	Sugarland	English, Joyce A.
*	123456789	2	7.5	ProductY	Sugarland	Wong, Franklin T.
	666884444	3	40.0	ProductZ	Houston	Narayan, Ramesh K.
*	666884444	3	40.0	ProductZ	Houston	Wong, Franklin T.
*	453453453	1	20.0	ProductX	Bellaire	Smith, John B.
	453453453	1	20.0	ProductX	Bellaire	English, Joyce A.
*	453453453	2	20.0	ProductY	Sugarland	Smith, John B.
	453453453	2	20.0	ProductY	Sugarland	English, Joyce A.
*	453453453	2	20.0	ProductY	Sugarland	Wong, Franklin T.
*	333445555	2	10.0	ProductY	Sugarland	Smith, John B.
*	333445555	2	10.0	ProductY	Sugarland	English, Joyce A.
	333445555	2	10.0	ProductY	Sugarland	Wong, Franklin T.
*	333445555	3	10.0	ProductZ	Houston	Narayan, Ramesh K.
	333445555	3	10.0	ProductZ	Houston	Wong, Franklin T.
	333445555	10	10.0	Computerization	Stafford	Wong, Franklin T.
*	333445555	20	10.0	Reorganization	Houston	Narayan, Ramesh K.
	333445555	20	10.0	Reorganization	Houston	Wong, Franklin T.

*
*
*

Additional tuples that were not there in Emp_proj is here, they are called spurious tuples

2. Functional Dependencies

Functional dependencies (FDs)

- Are used to specify *formal measures* of the "goodness" of relational designs
- And keys are used to define **normal forms** for relations
- Are **constraints** that are derived from the *meaning* and *interrelationships* of the data attributes

A set of attributes X *functionally determines* a set of attributes Y if the value of X determines a unique value for Y

2.1 Defining Functional Dependencies

X → Y holds 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]

X → Y in R specifies a *constraint* on all relation instances r(R)

Written as X → Y; can be displayed graphically on a relation schema as in Figures; denoted by the arrow →

FDs are derived from the real-world constraints on the attributes

Examples of FD constraints (1)

Social security number determines employee name

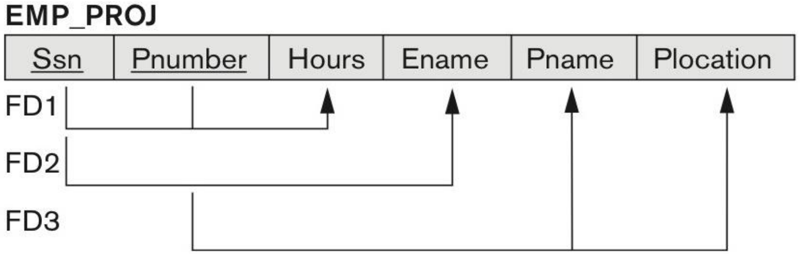
SSN → ENAME

Project number determines project name and location

PNUMBER → {PNAME, PLOCATION}

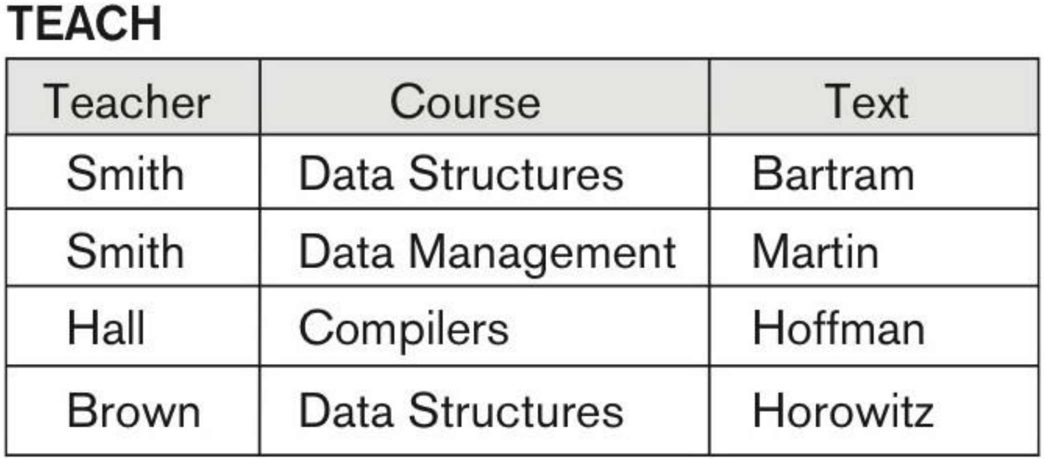
Employee ssn and project number determines the hours per week that the employee works on the project

{SSN, PNUMBER} → HOURS



Ruling Out FDs

Note that given the state of the TEACH relation, we can say that the FD: Text → Course may exist. However, the FDs Teacher → Course, Teacher → Text and Course → Text are ruled out.



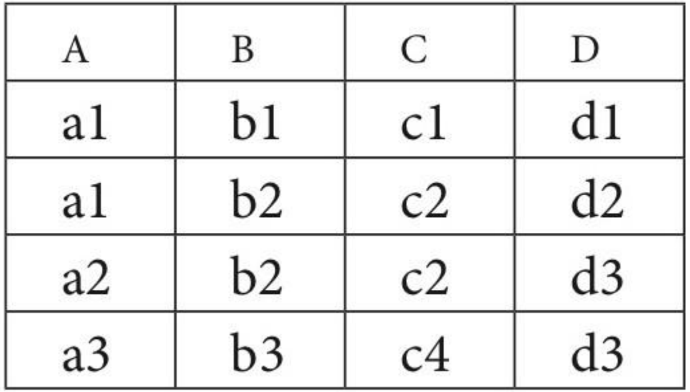
TEACH

Teacher	Course	Text
Smith	Data Structures	Bartram
Smith	Data Management	Martin
Hall	Compilers	Hoffman
Brown	Data Structures	Horowitz

What FDs may exist?

A relation *R*(A, B, C, D) with its extension.

Which FDs *may exist* in this relation?



A	B	C	D
a1	b1	c1	d1
a1	b2	c2	d2
a2	b2	c2	d3
a3	b3	c4	d3

B → C; C → B; {A,B} → C; {A,B} → D; {C,D}→ B

How about A → B? B→ A? D → C?

3.2 Practical Use of Normal Forms

Normalization 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 and BCNF. 4NF rarely used in practice.)

Denormalization:

The process of storing the join of higher normal form relations as a base relation—which is in a lower normal form

3.4 First Normal Form

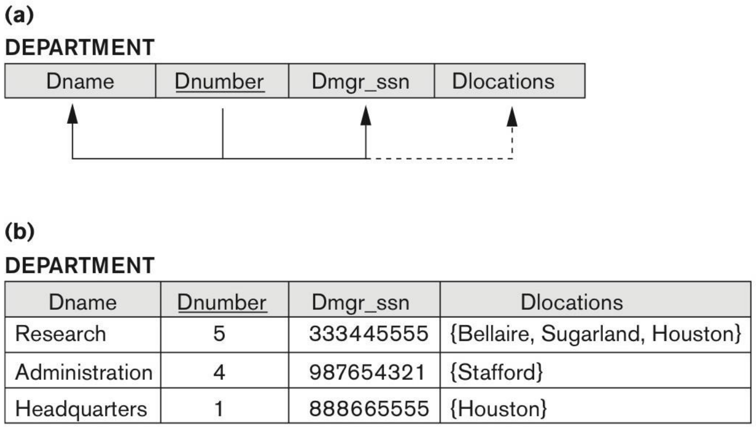
Disallows

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

Considered to be part of the definition of a relation

Most RDBMSs allow only those relations to be defined that are in First Normal Form

Normalization into 1NF



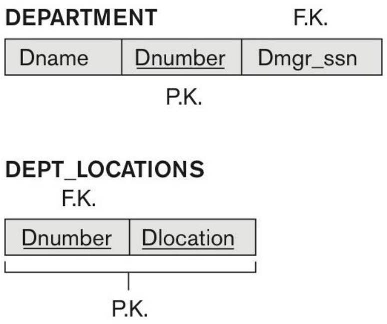
Dname	Dnumber	Dmgr_ssn	Dlocations
Research	5	333445555	{Bellaire, Sugarland, Houston}
Administration	4	987654321	{Stafford}
Headquarters	1	888665555	{Houston}

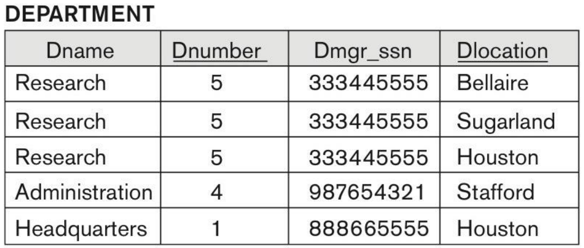
Figure 14.9
Normalization into 1NF. (a) A relation schema that is not in 1NF. (b) Sample state of relation DEPARTMENT

Ways to make it make it 1NF?

1NF

2 = Redundancy, Dnumber & Dlocation primary key





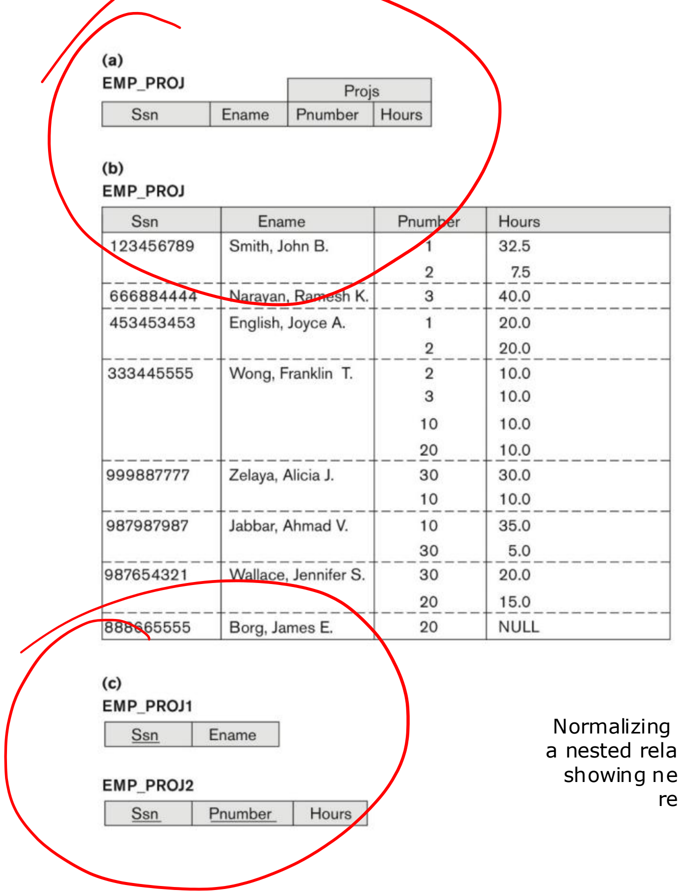
DEPARTMENT

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

1 = Two 1NF relations

3 = If the maximum number of values (n) for location is known, replace it with n attributes
e.g. Only 3 locations for the company – Dlocation1, Dlocation2, Dlocation3
Introducing NULL if most departments have fewer than 3 locations
Hard to query, e.g. List the departments that have ‘Bellaire’ as one of the locations
1st option is commonly used one

Normalizing 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, 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

Ssn is the primary key, Pnumber is the partial key

Remove the nested relation attributes into a new relation and propagate primary key

This idea can be applied recursively to a relation with multiple-level nesting to unnest

BLOB, CLOB – atomic, single-valued so 1NF

Figure 14.10

Normalizing nested relations into 1NF. (a) Schema of the EMP_PROJ relation with a nested relation attribute PROJS. (b) Sample extension of the EMP_PROJ relation showing nested relations within each tuple. (c) Decomposition of EMP_PROJ into relations EMP_PROJ1 and EMP_PROJ2 by propagating the primary key.

3.5 Second Normal Form (1)

Uses the concepts of **FDs, primary key**

Definitions

- **Prime attribute:** An 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

Examples:

- {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 a partial dependency) since SSN -> ENAME also holds

Second Normal Form (2)

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

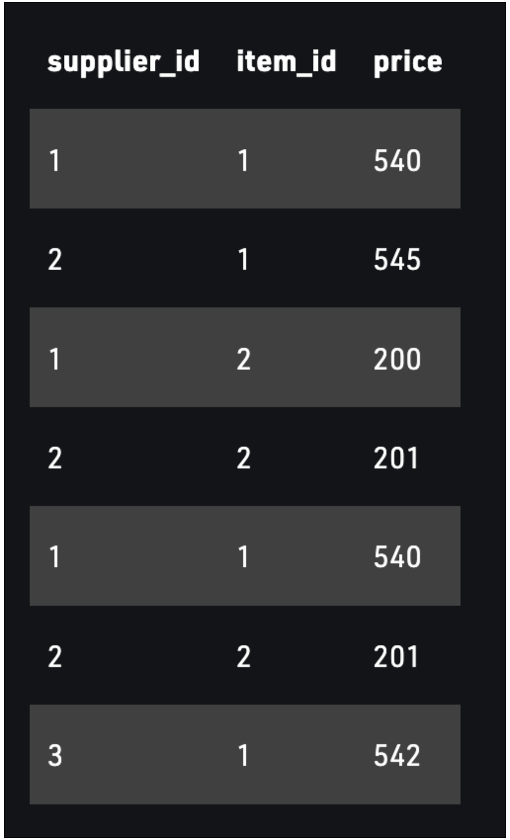
R can be decomposed into 2NF relations via the process of 2NF normalization or “second normalization”

This Lecture

Fully functional dependency

If X and Y are an attribute set of a relation, Y is fully functional dependent on X, if Y is functionally dependent on X but not on any proper subset of X.

e.g. In the relation ABC->D, attribute D is fully functionally dependent on ABC and not on any proper subset of ABC. That means that subsets of ABC like AB, BC, A, B, etc cannot determine D.



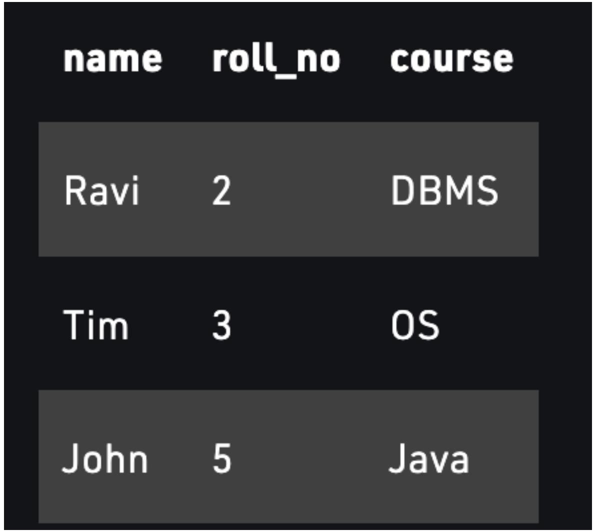
supplier_id	item_id	price
1	1	540
2	1	545
1	2	200
2	2	201
1	1	540
2	2	201
3	1	542

{ supplier_id , item_id } -> price

https://www.geeksforgeeks.org/differentiate-between-partial-dependency-and-fully-functional-dependency/

Partial dependency

A functional dependency X->Y is a partial dependency if Y is functionally dependent on X and Y can be determined by any proper subset of X.

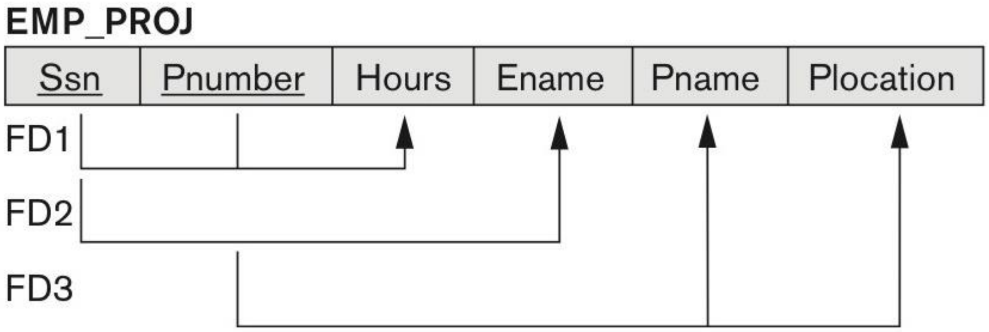


name	roll_no	course
Ravi	2	DBMS
Tim	3	OS
John	5	Java

{name} → course
{roll_no} → course

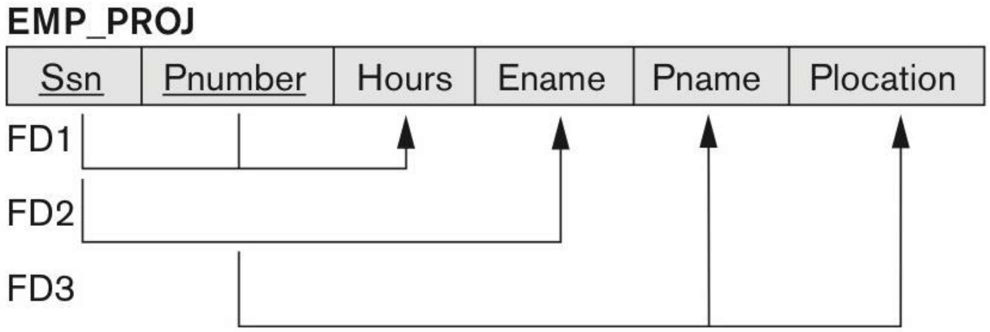
https://www.geeksforgeeks.org/differentiate-between-partial-dependency-and-fully-functional-dependency/

2NF



Any violation?

2NF



EMP_PROJ

<u>Ssn</u>	<u>Pnumber</u>	Hours	Ename	Pname	Plocation

Any violation?

FD2, FD3 violates 2NF

{ssn} → {Ename}, {pnumber} → {pname, plocation} & {ssn, pnumber} are primary keys; not fully functionally dependent

Normalizing into 2NF

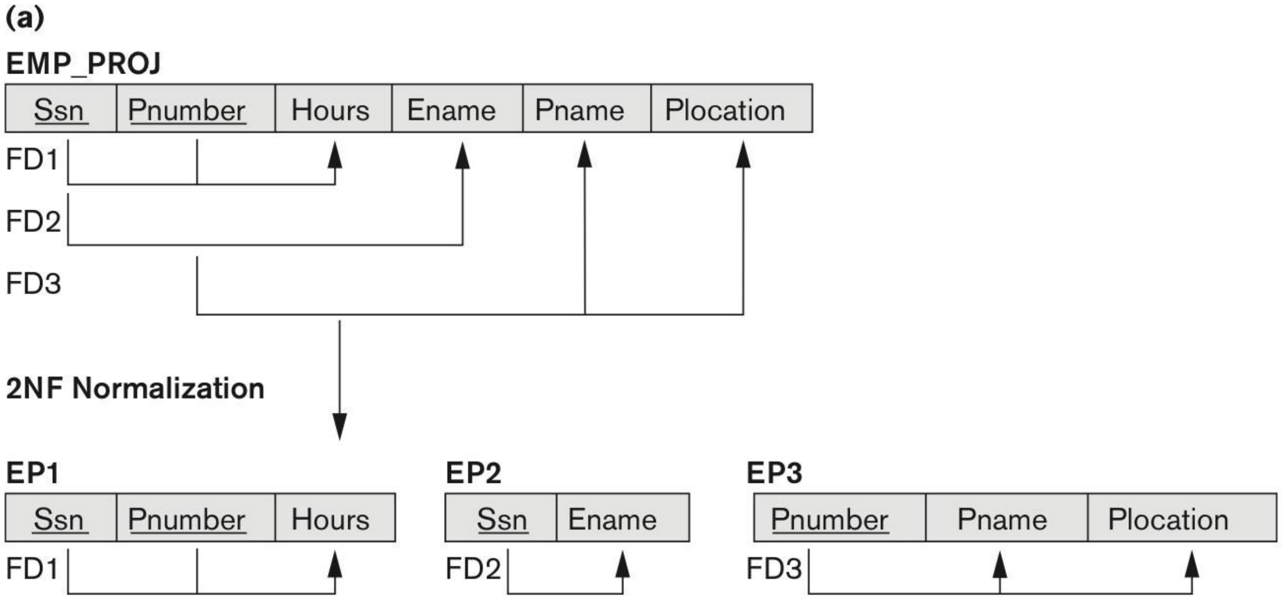


Figure 14.11
Normalizing into 2NF and 3NF.
(a) Normalizing EMP_PROJ into
2NF relations.

EP1, EP2, EP3 are fully functionally dependent

Third normal form

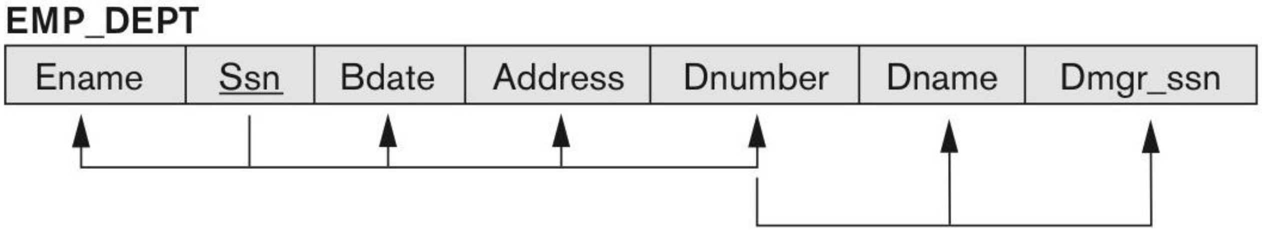
Transitive Dependency

$X \rightarrow Y$ in R is transitive dependency, if there exists a set of attributes Z in R that is neither a candidate key nor a subset of any key of R and both $X \rightarrow Z$ & $Z \rightarrow Y$ hold.

Third normal form

Transitive Dependency

X → Y in R is transitive dependency, if there exists a set of attributes Z in R that is neither a candidate key nor a subset of any key of R and both X → Z & Z → Y hold.



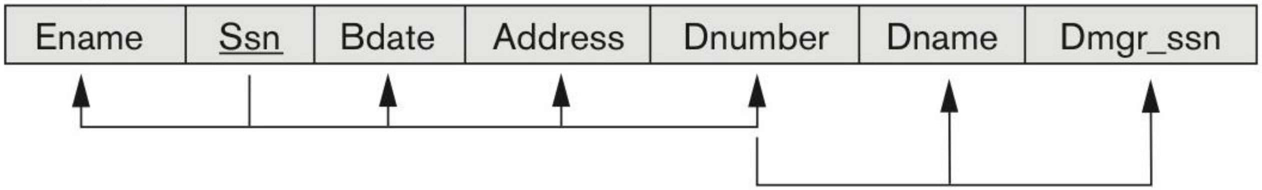
Any transitivity?

Third normal form

Transitive Dependency

X → Y in R is transitive dependency, if there exists a set of attributes Z in R that is neither a candidate key no~~t~~r a subset of any key of R and both X → Z & Z → Y hold.

EMP_DEPT



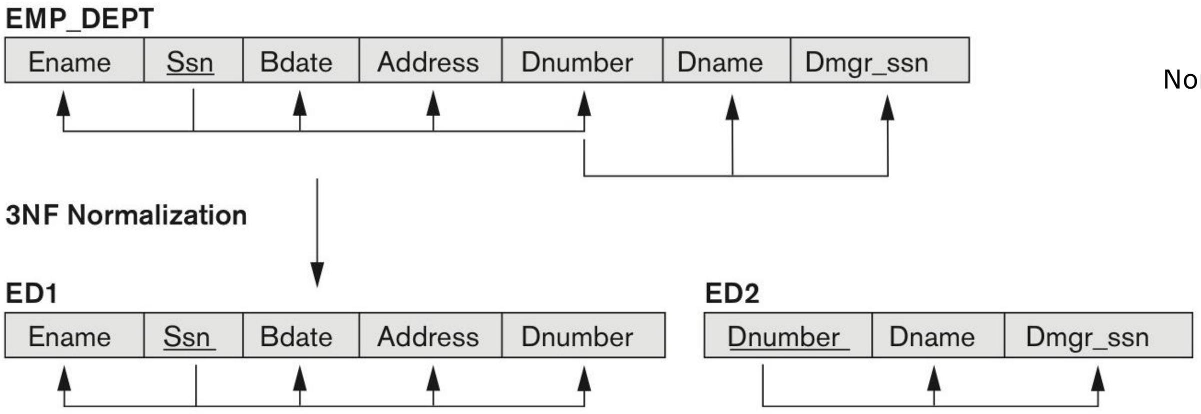
Any transitivity?

Ssn → dmgr_ssn is transitive through dnumber

Both ssn → dnumber & dnumber → dmgr_ssn hold & dnumber is neither a key nor a subset of a key

Third normal form

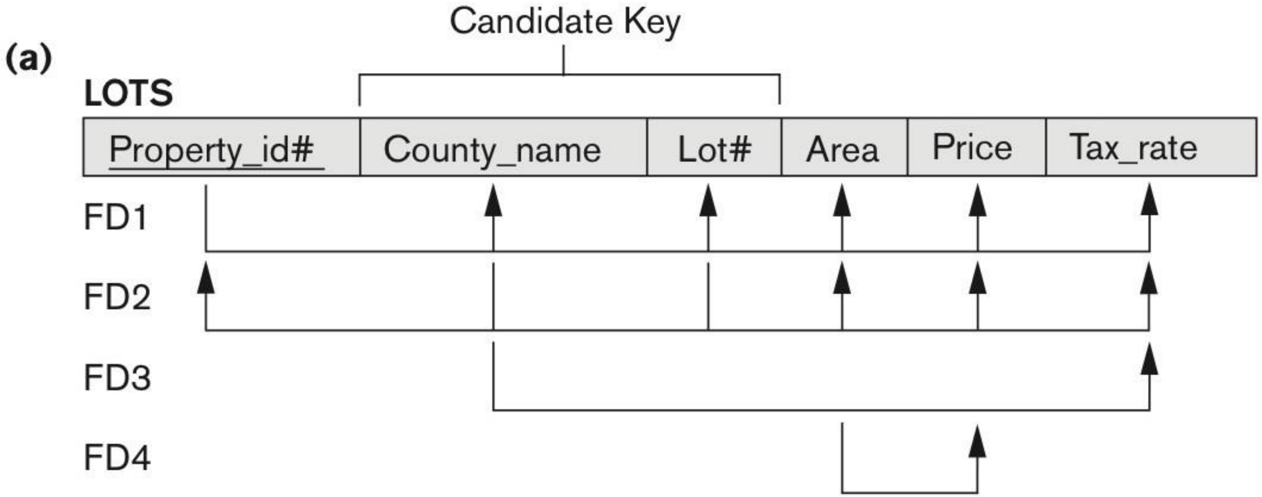
R is in 3NF if it satisfies 2NF and no nonprime attribute of R is transitively dependent on the primary key



Normalizing EMP_DEPT into 3NF relations.

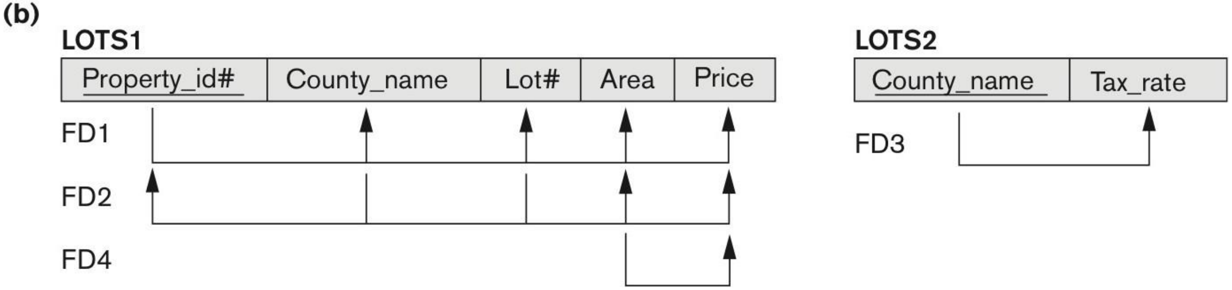
ED1 & ED2 represent independent facts about employees & departments

Figure 14.12a Normalization into 2NF and 3NF. The LOTS relation with its functional dependencies FD1 through FD4.



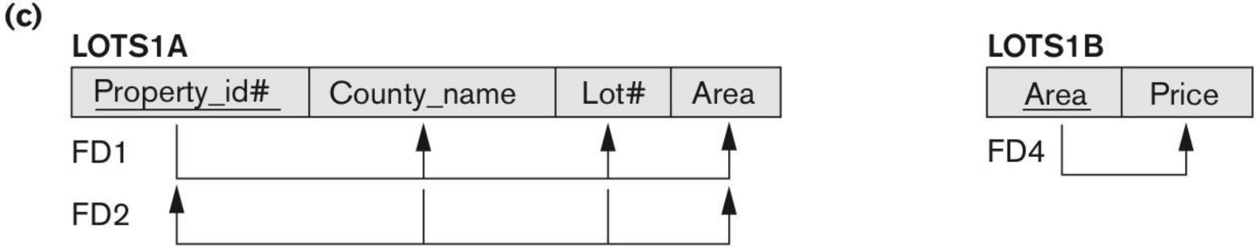
continued on next slide

Figure 14.12b Normalization into 2NF and 3NF. Decomposing into the 2NF relations LOTS1 and LOTS2.



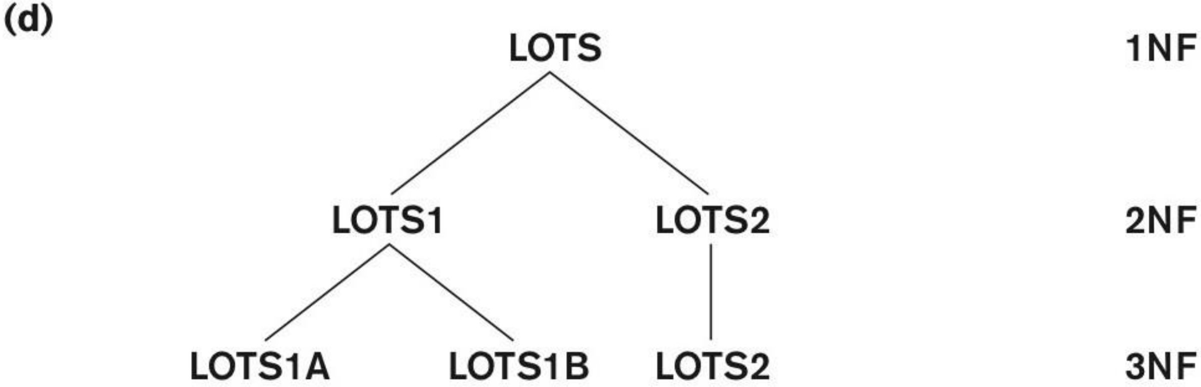
continued on next slide

Figure 14.12c Normalization into 2NF and 3NF. Decomposing LOTS1 into the 3NF relations LOTS1A and LOTS1B.



continued on next slide

Figure 14.12d Normalization into 2NF and 3NF. Progressive normalization of LOTS into a 3NF design.



5. 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

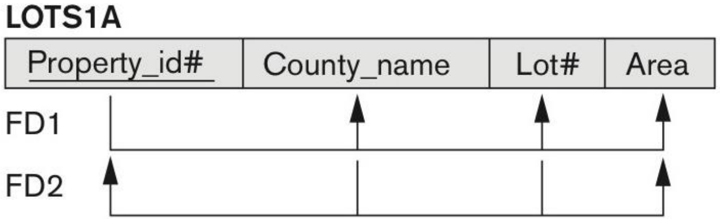
- Every 2NF relation is in 1NF
- Every 3NF relation is in 2NF
- Every BCNF relation is in 3NF

There exist relations that are in 3NF but not in BCNF

Hence BCNF is considered a stronger form of 3NF

The goal is to have each relation in BCNF (or 3NF)

Figure 14.13 Boyce-Codd normal form



Is this in BCNF?

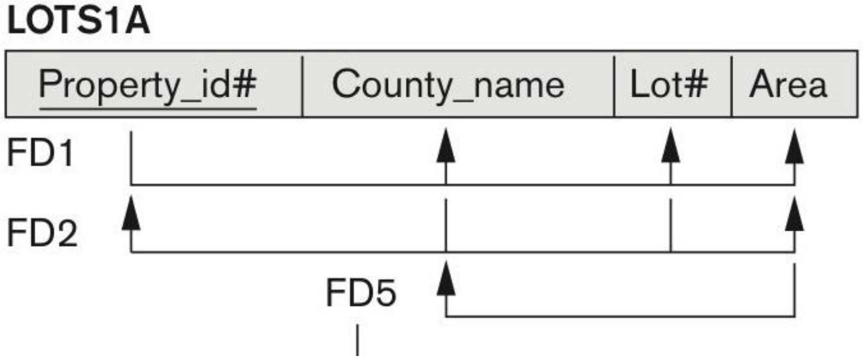
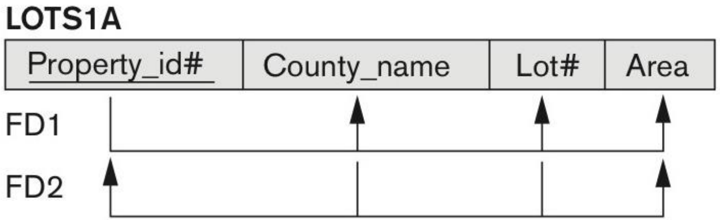
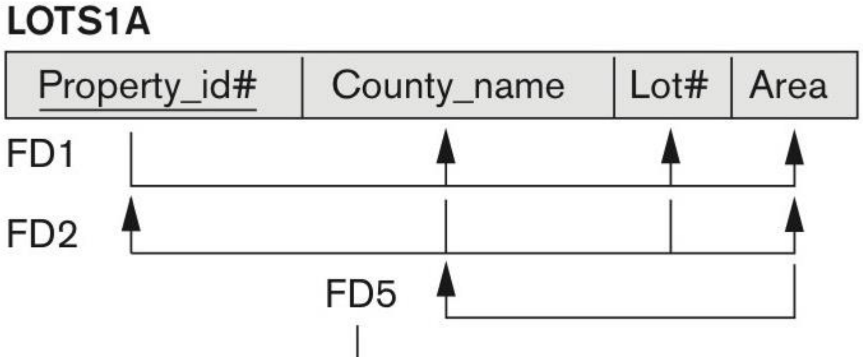


Figure 14.13 Boyce-Codd normal form

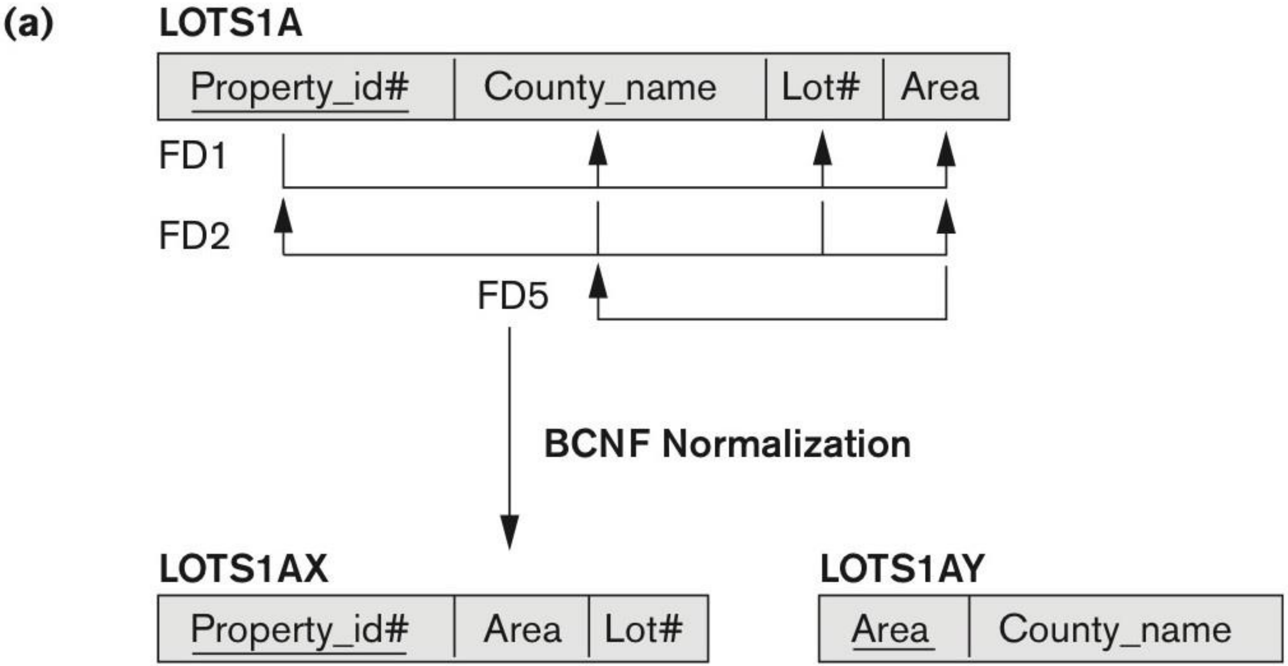




Is this in BCNF?

Area is not a superkey of LOTS1A

Figure 14.13 Boyce-Codd normal form



LOTS1A

Property_id#	County_name	Lot#	Area

LOTS1AX

Property_id#	Area	Lot#

LOTS1AY

Area	County_name

Normalization

https://www.youtube.com/watch?v=ABwD8IYByfk

Recap

Database & Database users

Database system concepts & architecture

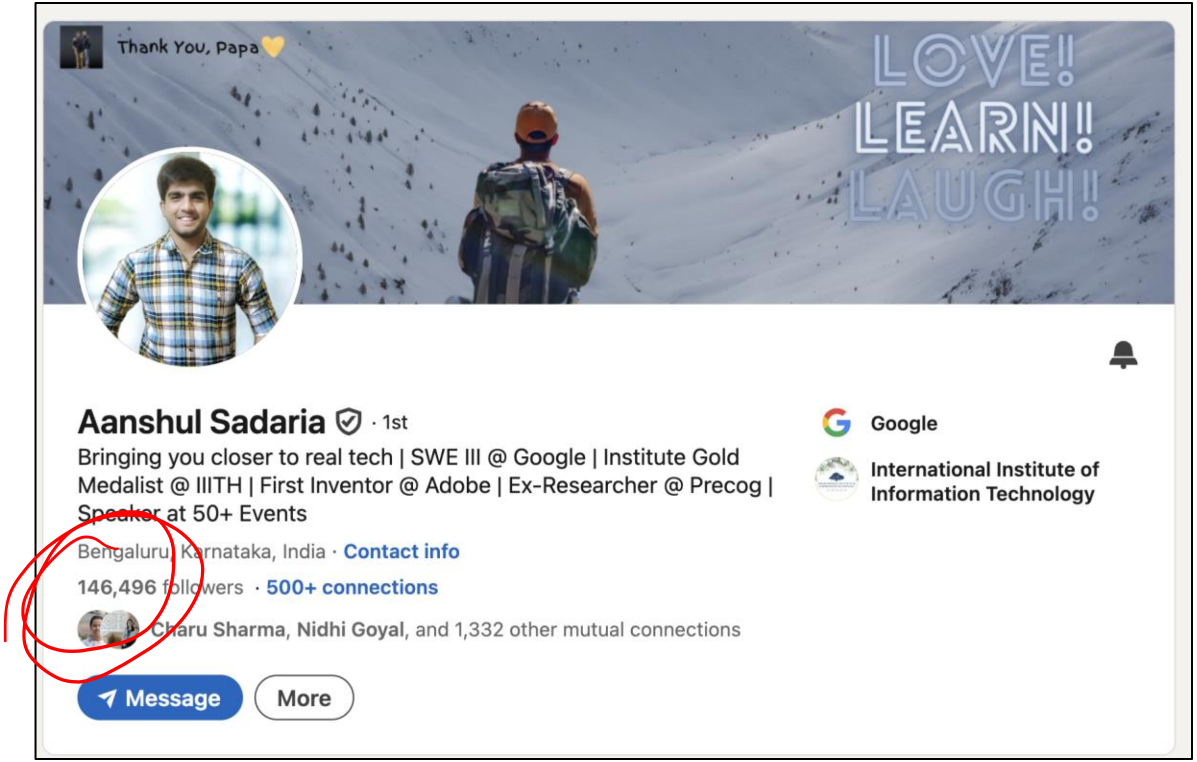
Data modeling / ER

Relational data model

SQL - Basic, complex, ..

Functional dependencies & normalization

Monday plan



Aanshul Sadaria · 1st

Bringing you closer to real tech | SWE III @ Google | Institute Gold Medalist @ IIITH | First Inventor @ Adobe | Ex-Researcher @ Precog | Speaker at 50+ Events

Bengaluru, Karnataka, India · Contact info

146,496 followers · 500+ connections

Charu Sharma, Nidhi Goyal, and 1,332 other mutual connections

Message More

Google

International Institute of Information Technology

Quiz

Search for: Ponnurangam Kumaraguru

https://www.linkedin.com/in/ponguru/

@PK.PROFGIRI

https://twitter.com/ponguru

Interested in working with us?

Look for an email in Jan 1st week after start of classes

NLP, Applied Machine Learning, Responsible & Safe AI, Social Network Analysis

Process: Apply (SOP, CV, etc.) – Task – Technical Interview

All done by mid sem

Bibliography / Acknowledgements

Instructor materials from Elmasri & Navathe 7e pk.profgiri

Ponnurangam.kumaraguru

/in/ponguru

ponguru

pk.guru@iiit.ac.in

Thank you
for attending
the class!!!