

02 Database normalization

Database normalization

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015

1

02 Database normalization

First normal form (1NF)

Relational schema is in 1NF if all occurrences of rows in the respective relational table contain the same number of fields and include the atomic values only, (there is no repeating fields and groups)

e#	name	car used
950001	Peter	Toyota, PKR234, Ford, WER545
932345	Paul	Honda, RTQ456
960020	Joan	Holden, KLR197, Holden, KLR567

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015

2

02 Database normalization

Keys

Superkey

Superkey is a nonempty subset X of relational schema $R = (A_1, \dots, A_n)$ such that for any two rows t_1, t_2 in a relational table defined over R $t_1[X] \neq t_2[X]$

If X is a superkey in R then $X \rightarrow (A_1, \dots, A_n)$

Minimal key

Minimal key is a superkey K with an additional property such that removal of any attribute from K will cause K not to be a superkey

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015

3

02 Database normalization

Keys

Primary key

Primary key is an arbitrarily selected minimal key

Candidate key

Candidate key is any other minimal key which is not primary

important key def here

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015

4

02 Database normalization

Functional dependencies and keys

Let $R = (A_1, \dots, A_n)$ be a relational schema (a header of relational table) and let X, Y be nonempty subsets of R such that $X \cup Y = R$

- (1) If $X \rightarrow Y$ is valid in R then X is a key (!!!)
- (2) If X is a key then $X \rightarrow Y$ is valid in R

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015

5

02 Database normalization

Attributes

Prime attribute

Prime attribute is an attribute from relational schema R which is a member of at least one candidate key in R

Nonprime attribute

Nonprime attribute is an attribute which is a not prime

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015

6

02 Database normalization

Full and partial functional dependencies

Full functional dependency

Full functional dependency is a functional dependency $X \rightarrow Y$ such that removal of any attribute A from X causes that $(X-A) \not\rightarrow Y$

Partial functional dependency

Partial functional dependency is a functional dependency which is not full

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015

7

02 Database normalization

Second normal form (2NF)

Relational schema R is in 2NF if every nonprime attribute A in R is fully functionally dependent on a primary key of schema R

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015

8

02 Database normalization

Second normal form (2NF)

Inventory

part	quantity	warehouse	warehouse-address
------	----------	-----------	-------------------

$\text{warehouse} \rightarrow \text{warehouse-address}$

$\text{part, warehouse} \rightarrow \text{quantity}$

$\text{part, warehouse} \rightarrow \text{warehouse-address}$

$\text{part, warehouse} \rightarrow \text{warehouse-address, quantity}$

Minimal key = (part, warehouse)

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015

9

02 Database normalization

Second normal form (2NF)

Inventory

part	quantity	warehouse	warehouse-address
------	----------	-----------	-------------------

$\text{warehouse} \rightarrow \text{warehouse-address}$

$\text{part, warehouse} \rightarrow \text{quantity}$

Minimal key = (part, warehouse)

Schema **Inventory** is **not in 2NF** because nonprime attribute **warehouse-address** depends on a part (warehouse) of a key (part, warehouse)

Functional dependency which "violates 2NF" is **warehouse \rightarrow warehouse-address**

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015

10

02 Database normalization

Second normal form (2NF)

Inventory

part	quantity	warehouse	warehouse-address
------	----------	-----------	-------------------

$\text{warehouse} \rightarrow \text{warehouse-address}$

$\text{part, warehouse} \rightarrow \text{quantity}$

Minimal key = (part, warehouse)

If all minimal keys in a schema consist of only one attribute (single attribute keys) then such schema is always in 2NF

This is because any nonprime attribute in the schema does not depend on a part of a key (because each key consists of one attribute only !)

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015

11

02 Database normalization

Second normal form (2NF)

Inventory

part	quantity	warehouse	warehouse-address
------	----------	-----------	-------------------

Schema **Inventory** should be decomposed into the following schemas

Store

part	quantity	warehouse
------	----------	-----------

Location

warehouse	warehouse-address
-----------	-------------------

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015

12

02 Database normalization

Second normal form (2NF)

Store

part quantity warehouse

 $\text{part, warehouse} \rightarrow \text{quantity}$

Minimal key = (part, warehouse)

Schema **Store** is in 2NF because no nonprime attributes (quantity) depends on a part of a key (part, warehouse)

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 13

02 Database normalization

Second normal form (2NF)

Location

warehouse warehouse-address

 $\text{warehouse} \rightarrow \text{warehouse-address}$

Minimal key = (warehouse)

Schema **Location** is in 2NF because no nonprime attributes (warehouse-address) depends on a part of a key (warehouse)

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 14

02 Database normalization

Second normal form (2NF)

Location

warehouse warehouse-address

 $\text{warehouse} \rightarrow \text{warehouse-address}$

Minimal key = (warehouse)

Every relational schema, which consists of at most 2 attributes is always in 2NF

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 15

02 Database normalization

Transitive functional dependency

Functional dependency $X \rightarrow Y$ valid in a relational schema **R** is a transitive functional dependency if there exists a nonempty subset **Z** of **R**, that is not a subset of any key in **R** and such that the functional dependencies $X \rightarrow Z$ and $Z \rightarrow Y$ are valid in **R**

We say that **Y** is transitively dependent on X in schema **R** if $X \rightarrow Y$ is valid in **R** and $X \rightarrow Y$ is a transitive functional dependency

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 16

02 Database normalization

Third normal form (3NF)

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

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 17

02 Database normalization

Third normal form (3NF)

Supplier

s# sname company-name city

 $s\# \rightarrow sname$ $s\# \rightarrow company-name$ $s\# \rightarrow city$ $company-name \rightarrow city$

Minimal key = (s#)

Schema **Supplier** is in 2NF because there are no nonprime attributes which are functionally dependent on a part of minimal key s#

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 18

02 Database normalization

Third normal form (3NF)

Supplier			
s#	sname	company-name	city

$s\# \rightarrow sname$
 $s\# \rightarrow company-name$
 $s\# \rightarrow city$
 $company-name \rightarrow city$
Minimal key = (s#)

Schema **Supplier** is not in 3NF because attribute **city** is transitively dependent on **s#**
 $s\# \rightarrow company-name \ \& \ company-name \rightarrow city$

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 19

02 Database normalization

Third normal form (3NF)

Supplier			
s#	sname	company-name	city

Schema **Supplier** should be decomposed into the following schemas

Supplier-new		
s#	sname	company-name

Company	
company-name	city

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 20

02 Database normalization

Third normal form (3NF)

Supplier-new		
s#	sname	company-name

$s\# \rightarrow sname$
 $s\# \rightarrow company-name$
Minimal key = (s#)

Schema **Supplier-new** is in 3NF because no attribute is transitively dependent on **s#**

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 21

02 Database normalization

Third normal form (3NF)

Company	
company-name	city

$company-name \rightarrow city$
Minimal key = (company-name)

Schema **Company** is in 3NF because no attribute is transitively dependent on **company-name**

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 22

02 Database normalization

Third normal form (3NF)

Company	
company-name	city

$company-name \rightarrow city$

Every relational schema, which consists of at most 2 attributes is always in 3NF

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 23

02 Database normalization

Third normal form (3NF)

Alternative definition
A relational schema **R** is in 3NF if whenever a functional dependency $X \rightarrow A$ is valid in **R** then either:

- (1) **X** is a superkey in **R**, or
- (2) **A** is a prime attribute in **R**

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 24

02 Database normalization

Third normal form (3NF)

Supplier

s#	sname	company-name	city
----	-------	--------------	------

 $s\# \rightarrow sname$ $s\# \rightarrow company-name$ $s\# \rightarrow city$ $company-name \rightarrow city$

Minimal key = (s#)

Schema Supplier is not in 3NF because

- (1) attribute company_name is not a superkey and
 (2) attribute city is not prime

© Janusz R. Getta

CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015

25

02 Database normalization

Third normal form (3NF)

Location

city	street	zip-code
------	--------	----------

 $city, street \rightarrow zip-code$ $zip-code \rightarrow city$ Minimal key₁ = (city, street)Minimal key₂ = (zip-code, street)

Schema Location is in 3NF because attribute city used on the right hand side of functional dependency $zip-code \rightarrow city$ is prime

© Janusz R. Getta

CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015

26

02 Database normalization

Third normal form (3NF)

Location

city	street	zip-code
NY	55	484
NY	56	484
LA	55	473
LA	56	473
LA	57	474

Repetition of [LA ... 473] and [NY ... 484] is forced by functional dependency $zip-code \rightarrow city$

© Janusz R. Getta

CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015

27

02 Database normalization

Boyce-Codd normal form (BCNF)

A relational schema R is in BCNF if whenever functional dependency $X \rightarrow A$ holds in R then:

- (1) X is a superkey in R

- ~~(2) or A is a prime attribute in R~~ (3NF only !!!)

© Janusz R. Getta

CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015

28

02 Database normalization

Boyce-Codd normal form (BCNF)

Location

city	street	zip-code
------	--------	----------

 $city, street \rightarrow zip-code$ $zip-code \rightarrow city$ Minimal key₁ = (city, street)Minimal key₂ = (zip-code, street)

Schema Location is not in BCNF because zip-code is not a superkey

 $zipcode \rightarrow city \ \& \ zip_code \not\rightarrow street$

© Janusz R. Getta

CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015

29

02 Database normalization

Boyce-Codd normal form (BCNF)

Location

city	street	zip-code
------	--------	----------

Schema Location should be decomposed into the following schemas

CS

street	zip-code
--------	----------

CZ

city	zip-code
------	----------

© Janusz R. Getta

CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015

30

02 Database normalization

Boyce-Codd normal form (BCNF)

CS

street zip-code

Minimal key₂ = (zip-code, street)

Schema **CS** is in BCNF because does not exist a functional dependency whose left hand side is not a superkey

© Janusz R. Getta

CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015

31

02 Database normalization

Boyce-Codd normal form (BCNF)

CZ

city zip-code

zip-code → city

Minimal key = (zip-code)

Schema **CZ** is in BCNF because zip-code is a superkey

© Janusz R. Getta

CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015

32

02 Database normalization

Boyce-Codd normal form (BCNF)

CS

street zip-code

Minimal key = (zip-code, street)

CZ

city zip-code

zip-code → city

Minimal key = (zip-code)

Every relational schema, which consists of at most 2 attributes is always in BCNF

© Janusz R. Getta

CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015

33

02 Database normalization

Boyce-Codd normal form (BCNF)

CS

street zip-code

Minimal key = (zip-code, street)

CZ

city zip-code

zip-code → city

Minimal key = (zip-code)

Normalization to BCNF costs a functional dependency
city, street → zip-code

© Janusz R. Getta

CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015

34

02 Database normalization

Normalization of relational schemas

Given a relational schema $R = \{ A_1, \dots, A_n \}$

Identify all functional dependencies

Use functional dependencies to derive all minimal keys

Use functional dependencies and minimal keys to identify the highest normal form satisfied by R

Decompose R into the schemas in BCNF (3NF)

© Janusz R. Getta

CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015

35

02 Database normalization

Normalization of relational schemas

Shipment

s# city status p# quantity

s# → city

s# → status

city → status

s#, p# → quantity

s#, p# → city

s#, p# → status

Minimal key = (s#, p#)

Schema **Shipment** is in 1NF

© Janusz R. Getta

CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015

36

02 Database normalization

Normalization of relational schemas

Shipment

s# city status p# quantity

 $s\# \rightarrow city$ $s\# \rightarrow status$ $city \rightarrow status$ $s\#, p\# \rightarrow quantity$ $s\#, p\# \rightarrow city$ $s\#, p\# \rightarrow status$ Schema **Shipment** is not in 2NFNonprime attribute **city** depends on a part of a key ($s\#, p\#$)Minimal key = ($s\#, p\#$)

© Janusz R. Getta

CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015

37

02 Database normalization

Normalization of relational schemas

Shipment

s# city status p# quantity

Schema **Shipment** should be decomposed into the following schemas

SP

s# p# quantity

Supplier

s# city status

Schema **SP** is in BCNF $s\#, p\# \rightarrow quantity$ Schema **Supplier** is not in 3NF(& BCNF) $s\# \rightarrow city$ $city \rightarrow status$ $s\# \rightarrow status$

© Janusz R. Getta

CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015

38

02 Database normalization

Normalization of relational schemas

Supplier

s# city status

Schema **Supplier** should be decomposed into the following schemas

Supplier'

s# city

Schema **Supplier'** is in BCNF $s\# \rightarrow city$

Location

city status

Schema **Location** is in BCNF $city \rightarrow status$

© Janusz R. Getta

CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015

39

02 Database normalization

Normalization of relational schemas

Supplier

s# city status

Schema **Supplier** may be alternatively decomposed into the following schemas

Supplier'

s# city

Schema **Supplier'** is in BCNF $s\# \rightarrow city$

Supplier''

s# status

Schema **Supplier''** is in BCNF $s\# \rightarrow status$

© Janusz R. Getta

CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015

40

02 Database normalization

References

Elmasri R., Navathe S. B., *Fundamentals of Database Systems*, chapters 10.3,10.4,10.5R. Ramakrishnan, J. Gehrke *Database Management Systems*, chapters 19.2, 19.3

© Janusz R. Getta

CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015

41