Bachelor of Science (Honours)
in
Data Science and Artificial Intelligence

DA201: Relational Database Management Systems

Relational Database Design
– Normalization

by
Dr. Manas Khatua
Dept. of Computer Science and Engineering
Indian Institute of Technology Guwahati
Email: manaskhatua@iitg.ac.in

Learning Objectives

1) Normal Forms

2) Decomposition using Functional Dependencies

3) Decomposition Using Multivalued Dependencies

Lets have recap …

- Relational Data Model: This conceptual model uses *tables* (or *relations*) to represent both data and the relationships among those data
- Data Redundancy: It is the repetition of same data in multiple places within a relation or database
- Functional Dependency: FD is a constraint between two sets of attributes from the database relation
- Closure of a Set of FDs: The set of all FDs logically implied by the given set of FDs F is the closure of F (denoted by F^+)
- Closure of Attribute Sets: The closure of attribute set α under F (denoted by α^+) is the set of attributes that are functionally determined by α under F
- Lossless Decomposition: A decomposition is lossless if there is no loss of information by replacing relation R with the two relation schemas R_1 and R_2
- Dependency Preserving: A decomposition that makes it computationally hard to enforce FDs is said to be NOT dependency preserving

Normalization Theory

❖ The method for designing good relational database is to use a process called **Normalization**.

❖ The **purpose** of Normalization:

- Minimizing redundancy
- Minimizing inconsistency

❖ The **approach** of Normalization:

- Certify whether a particular relation R is in "good form"
  - There exist many "good forms" called **normal forms (NF)**.

- In the case that a relation R is not in "good form", decompose it into set of relations $\{R_1, R_2, \ldots, R_n\}$ such that
  - Each relation is in "good form"
  - The decomposition is a lossless decomposition
  - Preferably, all the functional dependencies are preserved.

Normal Forms

- Different Normal Forms:
  - Based on FDs
    - First normal form (1NF)
    - Second normal form (2NF)
    - Third normal form (3NF)
    - Boyce-Codd normal form (BCNF)
  - Based on Multivalued FDs
    - Fourth normal form (4NF)
  - Based on Join FDs
    - Fifth normal form (5NF)

Normal Forms

- Different Normal Forms:
  - Based on FDs
    - First normal form (1NF)
    - Second normal form (2NF)
    - Third normal form (3NF)
    - Boyce-Codd normal form (BCNF)
  - Based on Multivalued FDs
    - Fourth normal form (4NF)
  - Based on Join FDs
    - Fifth normal form (5NF)

The normal form (NF) of a relation indicates the **highest degree** to which it has been normalized.

Note: Normal forms only do not guarantee a good database design. It only indicates the present status of the design.

Lossless Decomposition

We can use FDs to show when certain decomposition are lossless.

- For the case of $R = (R_1, R_2)$, we require that for all possible relations r on schema R

$$r = \Pi_{R1}(r) \bowtie \Pi_{R2}(r)$$

A decomposition of R into R_1 & R_2 is lossless if at least one of the following dependencies is in F^+:

- $R_1 \cap R_2 \rightarrow R_1$
- $R_1 \cap R_2 \rightarrow R_2$

- *Example:*

  $R = (A, B, C)$
  $F = \{A \rightarrow B, B \rightarrow C)$

Decomposition:
$R_1 = (A, B), \quad R_2 = (A, C)$

It is lossless as,
$R_1 \cap R_2 = \{A\}$ and $A \rightarrow AB$

Decomposition:
$R_1 = (A, C), \quad R_2 = (B, C)$

It is lossy as,
$R_1 \cap R_2 = \{C\}$ and $C \rightarrow ??$

Dependency Preservation

- Let F_i be the set of dependencies in F^+ that include only attributes in relation R_i.
  - A decomposition is **dependency preserving**, if and only if

$$(F_1 \cup F_2 \cup \ldots \cup F_n)^+ = F^+$$

- *Example:*
  $R = (A, B, C)$
  $F = \{A \to B$
  $B \to C\}$
  Candidate Key = $\{A\}$

  Decomposition
  $R_1 = (A, B), \quad R_2 = (B, C)$

  Corresponding FDs:
  $F_1 = \{A \to B\}, \quad F_2 = \{B \to C\}$

  So, FDs are **preserved**.

- *Example:*
  $R = (A, B, C)$
  $F = \{AB \to C$
  $C \to B\}$
  Candidate Keys = $\{AB\}$, {AC}

  Decomposition
  $R_1 = (A, B), \quad R_2 = (B, C)$

  Corresponding FDs:
  $F_1 = \{ \}, \quad F_2 = \{C \to B\}$

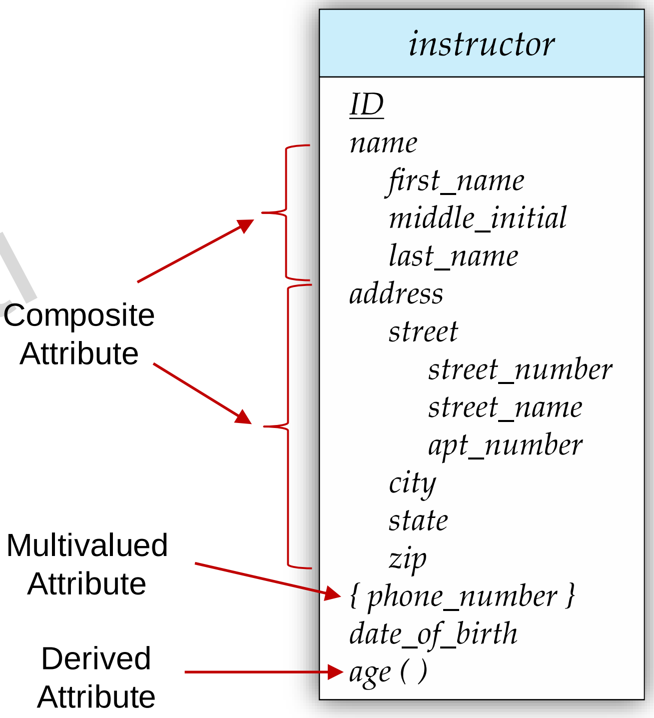
  So, FDs are **NOT preserved**.

First Normal Form (1NF)

- A relational schema *R* is in **1NF**
  - ✓ **if** the **domains** of all attributes of *R* are **atomic**.

- A domain is atomic
  - ✓ **if** its elements are considered to be indivisible units.
    - i.e., it disallow composite and multivalued attributes

instructor (*<u>ID</u>, name*{ }, *address*{ }, {*phone_number*}, *date_of_birth, age*())



First Normal Form (1NF)

- A relational schema *R* is in **1NF**
  - ✓ **if** the **domains** of all attributes of *R* are **atomic**.

- A domain is atomic
  - **if** its elements are considered to be indivisible units.
  - i.e., it disallow composite and multivalued attributes

instructor (<u>*ID*</u>, *name*{ }, *address*{ }, {*phone_number*}, *date_of_birth*, *age*())

Few techniques for decomposing a relation into 1NF:

1) Remove the attribute that violates 1NF and place it in a separate relation

2) If the maximum number of values for an attribute is *k*, then replace the attribute by *k* atomic attributes

3) Few more …

Example of 1NF Decomposition

instructor (*ID*, *name*{ }, *address*{ }, {*phone_number*}, *date_of_birth*, *age*())

Example of 1NF Decomposition

instructor (*ID*, *name*{ }, *address*{ }, {*phone_number*}, *date_of_birth*, *age*())

first_name,
middle_name,
last_name

We should
drop it.

address (*adrs_id*, *street*, *city*, *state*, *zip*)

inst_phone (*ID*, *phone_number*)

- Decomposed Relations

instructor (*ID*, *first_name*, *middle_name*, *last_name*, *adrs_id*, *date_of_birth*)

address (*adrs_id*, *street*, *city*, *state*, *zip*)

inst_phone (*ID*, *phone_number*)

Second Normal Form (2NF)

- A relation schema *R* is in **2NF, if**
  - *R* is in **1NF**, and
  - every nonprime attribute *A* in *R* is **fully functionally dependent** on the candidate key of *R*.

Example:

emp_proj (*emp_id*, *proj_no*, *emp_name, salary, proj_name, building, hours*)

FDs:

$emp\_id \rightarrow emp\_name, salary$

$proj\_no \rightarrow proj\_name, building$

$emp\_id, proj\_no \rightarrow hours$

Partial functional dependency on candidate key

Full functional dependency on candidate key

So, the *emp_proj* relation is in 1NF but not in 2NF.

Example of 2NF Decomposition

Example:

emp_proj (*emp_id*, *proj_no*, *emp_name, salary, proj_name, building, hours*)

FDs:

emp_id → *emp_name, salary*
proj_no → *proj_name, building*
emp_id, proj_no → *hours*

- Decomposition Rule:
  - Decomposed to the relations in which nonprime attributes are associated only with the part of the primary key on which they are fully functionally dependent

Example of 2NF Decomposition

Example:

emp_proj (<u>*emp_id*</u>, <u>*proj_no*</u>, *emp_name, salary, proj_name, building, hours*)

FDs:

emp_id $\rightarrow$ *emp_name, salary*
proj_no $\rightarrow$ *proj_name, building*
emp_id, proj_no $\rightarrow$ *hours*

Decomposed Relations:

employee (<u>*emp_id*</u>, *emp_name, salary*) — *FDs: emp_id* $\rightarrow$ *emp_name, salary*

project (<u>*proj_no*</u>, *proj_name, building*) — *FDs: proj_no* $\rightarrow$ *proj_name, building*

emp_proj_hrs (<u>*emp_id*</u>, <u>*proj_no*</u>, *hours*) — *FDs: emp_id, proj_no* $\rightarrow$ *hours*

- Does it lossless decomposition? => YES
- Are the FDs preserved? => YES

Third Normal Form (3NF)

A relation schema *R* is in **3NF,** if

- *R* is in **2NF,** and
- Any nonprime attribute of *R* is **not transitively dependent** on the candidate key of *R*.

OR

A relation schema *R* is in **3NF,** if

for all $\alpha \rightarrow \beta$ in F^+ ,

at least one of the following holds:

- $\alpha \rightarrow \beta$ is trivial (i.e., $\beta \in \alpha$)
- α is a superkey for *R*
- Each attribute *A* in $\beta - \alpha$ is contained in a candidate key for *R.*

Third Normal Form (3NF)

Example:

in_dept (*ID*, *name, salary, dept_name, building, budget*)

FDs:

ID $\rightarrow$ *name, salary, dept_name*

dept_name $\rightarrow$ *building, budget*

Does this relation in 3NF?

ID	*name*	*salary*	*dept_name*	*building*	*budget*
22222	Einstein	95000	Physics	Watson	70000
12121	Wu	90000	Finance	Painter	120000
32343	El Said	60000	History	Painter	50000
45565	Katz	75000	Comp. Sci.	Taylor	100000
98345	Kim	80000	Elec. Eng.	Taylor	85000
76766	Crick	72000	Biology	Watson	90000
10101	Srinivasan	65000	Comp. Sci.	Taylor	100000
58583	Califieri	62000	History	Painter	50000
83821	Brandt	92000	Comp. Sci.	Taylor	100000
15151	Mozart	40000	Music	Packard	80000
33456	Gold	87000	Physics	Watson	70000
76543	Singh	80000	Finance	Painter	120000

Third Normal Form (3NF)

Example:

in_dept (*ID*, *name, salary, dept_name, building, budget*)

FDs:

ID $\rightarrow$ *name, salary, dept_name*

dept_name $\rightarrow$ *building, budget*

ID	*name*	*salary*	*dept_name*	*building*	*budget*
22222	Einstein	95000	Physics	Watson	70000
12121	Wu	90000	Finance	Painter	120000
32343	El Said	60000	History	Painter	50000
45565	Katz	75000	Comp. Sci.	Taylor	100000
98345	Kim	80000	Elec. Eng.	Taylor	85000
76766	Crick	72000	Biology	Watson	90000
10101	Srinivasan	65000	Comp. Sci.	Taylor	100000
58583	Califieri	62000	History	Painter	50000
83821	Brandt	92000	Comp. Sci.	Taylor	100000
15151	Mozart	40000	Music	Packard	80000
33456	Gold	87000	Physics	Watson	70000
76543	Singh	80000	Finance	Painter	120000

Does this relation in 3NF?

NO.

- In *in_dept* relation, transitive dependency exist, but no partial dependency on primary key.
- So it is in 2NF, but not in 3NF.

Decomposition rule:

- Set up a relation including the nonkey attribute(s) that functionally determine(s) other nonkey attribute(s).

Third Normal Form (3NF)

Example:

in_dept (*<u>ID</u>*, *name, salary, dept_name, building, budget*)

FDs:

ID → *name, salary, dept_name*

dept_name → *building, budget*

Decomposition:

Third Normal Form (3NF)

Example:

in_dept (*ID*, *name, salary, dept_name, building, budget*)

FDs:

ID → *name, salary, dept_name*

dept_name → *building, budget*

Decomposition:

- Relation 1: *departmet* (*dept_name*, *building*, *budget*)
- FDs: dept_name → building, budget

- Relation 2: *instructor* (*ID*, *name*, *salary*, *dept_name*)
- FDs: ID → name, salary, dept_name

Third Normal Form (3NF)

Example 2:

dept_advisor (*s_ID*, *i_ID*, *dept_name*)

FDs:

$i\_ID \rightarrow dept\_name$

$s\_ID, dept\_name \rightarrow i\_ID$

Candidate keys = {*s_ID*, *dept_name*}, {*s_ID*, *i_ID*}

Does this relation in 3NF?

YES

Explanation:

- {*s_ID*, *dept_name*} is a superkey
- $i\_ID \rightarrow dept\_name$ and *i_ID* is NOT a superkey, but:

  { *dept_name*} – {*i_ID* } = {*dept_name* } and

  dept_name is contained in a candidate key

Boyce-Codd Normal Form (BCNF)

- A relation schema *R* is in **BCNF** if
  - the relation *R* is **in 3NF**, and
  - whenever a nontrivial FD $\alpha \rightarrow \beta$ holds in *R*, then α is a superkey of R.

OR

- A relation schema *R* is in **BCNF** with respect to a set *F* of FDs if for all FDs in F^+ of the form

$$\alpha \rightarrow \beta$$

where $\alpha \subseteq R$ and $\beta \subseteq R$, at least one of the following holds:

- $\alpha \rightarrow \beta$ is trivial (i.e., $\beta \subseteq \alpha$)
- α is a superkey for *R*

Boyce-Codd Normal Form (BCNF)

Example:

dept_advisor (*s_ID*, *i_ID*, *dept_name*)

FDs:

$i\_ID \rightarrow dept\_name$

$s\_ID, dept\_name \rightarrow i\_ID$

Candidate keys = {*s_ID*, *dept_name*}, {*s_ID*, *i_ID*}

Does this relation in BCNF?

NO.

Explanation:

- {*s_ID*, *dept_name*} is a superkey
- But, only {*i_ID*} is NOT a superkey.

Assume:

- Instructor can be associated with only a single department.
- A student may have more than one advisor, but at most one corresponding to a given department.

Boyce-Codd Normal Form (BCNF)

Example:

dept_advisor (*s_ID, i_ID, dept_name*)

FDs:

$i\_ID \rightarrow dept\_name$

$s\_ID, dept\_name \rightarrow i\_ID$

Candidate keys = {*s_ID, dept_name*}, {*s_ID, i_ID*}

Decomposition:

Assume:

- Instructor can be associated with only a single department.
- A student may have more than one advisor, but at most one corresponding to a given department.

Boyce-Codd Normal Form (BCNF)

Example:

dept_advisor (*s_ID, i_ID, dept_name*)

FDs:
i_ID → *dept_name*
s_ID, dept_name → *i_ID*

Candidate keys = {*s_ID, dept_name*}, {*s_ID, i_ID*}

Decomposition:

- Relation 1: *stud_dept* (*s_ID*, *i_ID*)
- FDs: x

- Relation 2: *in_dept* (*i_ID*, *dept_name*)
- FDs: *i_ID* → *dept_name*

Note: All FDs are not preserved !

Assume:

- Instructor can be associated with only a single department.
- A student may have more than one advisor, but at most one corresponding to a given department.

Decomposition Example: 1NF – BCNF

class (*course_id*, *title*, *dept_name*, *credits*, *sec_id*, *semester*, *year*, *building*, *room_no*, *capacity*, {*day*, *time*})

FD1: *course_id, sec_id, semester*, *year* $\rightarrow$ *building*, *room_no*, *day*, *time*

FD2: *course_id* $\rightarrow$ *title*, *dept_name*, *credits*

FD3: *building*, *room_no* $\rightarrow$ *capacity*

FD4: *capacity* $\rightarrow$ *building*

Candidate Key = {*course_id, sec_id, semester*, *year*}

Decompose to 1NF:

Decomposition Example: 1NF – BCNF

class (*course_id*, *title*, *dept_name*, *credits*, *sec_id*, *semester*, *year*, *building*, *room_no*, *capacity*, {*day*, *time*})

FD1: *course_id, sec_id, semester*, *year* $\rightarrow$ *building*, *room_no*, *day*, *time*

FD2: *course_id* $\rightarrow$ *title*, *dept_name*, *credits*

FD3: *building*, *room_no* $\rightarrow$ *capacity*

FD4: *capacity* $\rightarrow$ *building*

Candidate Key = {*course_id, sec_id, semester*, *year*}

Decompose to 1NF:

class_1 (<u>*course_id*</u>, *title*, *dept_name*, *credits*, <u>*sec_id*</u>, <u>*semester*</u>, <u>*year*</u>, *building*, *room_no*, *capacity*, *slot_id*)

class_slot (<u>*slot_id*</u>, <u>*day*</u>, <u>*time*</u>)

FD1 to FD4: holds in relation *class_1*

Decomposition Example: 1NF – BCNF

class_1 (*<u>course_id</u>*, *title*, *dept_name*, *credits*, *<u>sec_id</u>*, *<u>semester</u>*, *<u>year</u>*, *building*, *room_no*, *capacity*, *slot_id*)

FD1: *course_id, sec_id, semester, year* $\rightarrow$ *building, room_no, slot_id*

FD2: *course_id* $\rightarrow$ *title, dept_name, credits*

FD3: *building, room_no* $\rightarrow$ *capacity*

FD4: *capacity* $\rightarrow$ *building*

class_slot (*<u>slot_id</u>*, *<u>day</u>*, *<u>time</u>*)

<u>Decompose to 2NF:</u>

Decomposition Example: 1NF – BCNF

class_1 (<u>*course_id*</u>, *title*, *dept_name*, *credits*, <u>*sec_id*</u>, <u>*semester*</u>, <u>*year*</u>, *building*, *room_no*, *capacity*, *slot_id*)

FD1: *course_id, sec_id, semester*, *year* $\rightarrow$ *building*, *room_no*, *slot_id*

FD2: *course_id* $\rightarrow$ *title*, *dept_name*, *credits*

FD3: *building*, *room_no* $\rightarrow$ *capacity*

FD4: *capacity* $\rightarrow$ *building*

class_slot (<u>*slot_id*</u>, <u>*day*</u>, <u>*time*</u>)

<u>Decompose to 2NF:</u>

class_1A (<u>*course_id*</u>, <u>*sec_id*</u>, <u>*semester*</u>, <u>*year*</u>, *building*, *room_no*, *capacity*, *slot_id*); holds FD1, FD3, FD4

course (<u>*course_id*</u>, *title*, *dept_name*, *credits*); holds FD2

class_slot (<u>*slot_id*</u>, <u>*day*</u>, <u>*time*</u>);

Decomposition Example: 1NF – BCNF

class_1A (*course_id*, *sec_id*, *semester*, *year*, *building*, *room_no*, *capacity*, *slot_id*)

class_slot (*slot_id*, *day*, *time*)

FD1: *course_id, sec_id, semester*, *year* → *building*, *room_no*, *slot_id*

FD3: *building*, *room_no* → *capacity*

FD4: *capacity* → *building*

course (*course_id*, *title*, *dept_name*, *credits*)

FD2: *course_id* → *title*, *dept_name*, *credits*

Decompose to 3NF:

Decomposition Example: 1NF – BCNF

class_1A (<u>*course_id*</u>, <u>*sec_id*</u>, <u>*semester*</u>, <u>*year*</u>, *building*, *room_no*, *capacity*, *slot_id*)

FD1: *course_id, sec_id, semester*, *year* $\rightarrow$ *building*, *room_no*, *slot_id*

FD3: *building*, *room_no* $\rightarrow$ *capacity*

FD4: *capacity* $\rightarrow$ *building*

class_slot (<u>*slot_id*</u>, <u>*day*</u>, <u>*time*</u>)

course (<u>*course_id*</u>, *title*, *dept_name*, *credits*)

FD2: *course_id* $\rightarrow$ *title*, *dept_name*, *credits*

<u>Decompose to 3NF:</u>

class_1A2 (<u>*course_id*</u>, <u>*sec_id*</u>, <u>*semester*</u>, <u>*year*</u>, *building*, *room_no*, *slot_id*); holds FD1

room_capacity (<u>*building*</u>, <u>*room_no*</u>, *capacity*); holds FD3, FD4

course (<u>*course_id*</u>, *title*, *dept_name*, *credits*); holds FD2

class_slot (<u>*slot_id*</u>, <u>*day*</u>, <u>*time*</u>)

Decomposition Example: 1NF – BCNF

class_1A2 (*<u>course_id</u>*, *<u>sec_id</u>*, *<u>semester</u>*, *<u>year</u>*, *building*, *room_no*, *slot_id*)

class_slot (*<u>slot_id</u>*, *<u>day</u>*, *<u>time</u>*)

FD1: *course_id, sec_id, semester*, *year* → *building*, *room_no*, *slot_id*

room_capacity (*<u>building</u>*, *<u>room_no</u>*, *capacity*)

FD3: *building*, *room_no* → *capacity*

FD4: *capacity* → *building*

course (*<u>course_id</u>*, *title*, *dept_name*, *credits*)

FD2: *course_id* → *title*, *dept_name*, *credits*

<u>Decompose to BCNF:</u>

Decomposition Example: 1NF – BCNF

class_1A2 (<u>*course_id*</u>, <u>*sec_id*</u>, <u>*semester*</u>, <u>*year*</u>, *building*, *room_no*, *slot_id*)

class_slot (<u>*slot_id*</u>, <u>*day*</u>, <u>*time*</u>)

FD1: *course_id, sec_id, semester*, *year* $\rightarrow$ *building*, *room_no*, *slot_id*

room_capacity (<u>*building*</u>, <u>*room_no*</u>, *capacity*)

FD3: *building*, *room_no* $\rightarrow$ *capacity*

FD4: *capacity* $\rightarrow$ *building*

course (<u>*course_id*</u>, *title*, *dept_name*, *credits*)

FD2: *course_id* $\rightarrow$ *title*, *dept_name*, *credits*

<u>Decompose to BCNF:</u>

class_1A2 (<u>*course_id*</u>, <u>*sec_id*</u>, <u>*semester*</u>, <u>*year*</u>, *building*, *room_no*, *slot_id*); holds FD1

course (<u>*course_id*</u>, *title*, *dept_name*, *credits*); holds FD2

class_slot (<u>*slot_id*</u>, <u>*day*</u>, <u>*time*</u>);

cap_building (<u>*capacity*</u>, *building*); holds FD4

room_capacity (<u>*room_no*</u>, <u>*capacity*</u>); **[FD3 is not preserved.]**

Multivalued Dependencies Definition

- Let R be a relation schema and let $\alpha \subseteq R$ and $\beta \subseteq R$.
- The **multivalued dependency**

$$\alpha \rightarrow\rightarrow \beta$$

holds on R **if** in any legal relation $r(R)$, for all pairs for tuples t_1 and t_2 in r such that $t_1[\alpha] = t_2[\alpha]$, there exist tuples t_3 and t_4 in r such that:

$$t_1[\alpha] = t_2[\alpha] = t_3[\alpha] = t_4[\alpha]$$

$$t_3[\beta] = t_1[\beta]$$
$$t_3[R - \alpha - \beta] = t_2[R - \alpha - \beta]$$

$$t_4[\beta] = t_2[\beta]$$
$$t_4[R - \alpha - \beta] = t_1[R - \alpha - \beta]$$

Tabular representation

	α	β	$R - \alpha - \beta$
t_1	$a_1 \ldots a_i$	$a_{i+1} \ldots a_j$	$a_{j+1} \ldots a_n$
t_2	$a_1 \ldots a_i$	$b_{i+1} \ldots b_j$	$b_{j+1} \ldots b_n$
t_3	$a_1 \ldots a_i$	$a_{i+1} \ldots a_j$	$b_{j+1} \ldots b_n$
t_4	$a_1 \ldots a_i$	$b_{i+1} \ldots b_j$	$a_{j+1} \ldots a_n$

Theory of Multivalued Dependency

- From the definition of multivalued dependency, we can derive the following rule:
  - If $\alpha \rightarrow \beta$, then $\alpha \rightarrow\rightarrow \beta$

  That is, every FD is also a multivalued dependency, but the vice-versa is not true always.

- The **closure** D^+ of D is the set of all functional and multivalued dependencies logically implied by D.
  - We can compute D^+ from D, using the formal definitions of functional dependencies and multivalued dependencies.

Fourth Normal Form (4NF)

- A relation schema R is in **4NF** with respect to a set D of functional and multivalued dependencies,

  if for all multivalued dependencies in D^+ of the form $\alpha \rightarrow\rightarrow \beta$, where $\alpha \subseteq R$ and $\beta \subseteq R$, at least one of the following hold:

  - $\alpha \rightarrow\rightarrow \beta$ is trivial (i.e., $\beta \subseteq \alpha$ or $\alpha \cup \beta = R$)
  - α is a superkey for schema R

- If a relation is in 4NF, then it is in BCNF

Fourth Normal Form (4NF)

- *Example*:
  - $R = (\underline{A}, B, \underline{C}, \underline{G}, H, I)$

    $F = \{ A \rightarrow\rightarrow B$

    $B \rightarrow\rightarrow HI$

    $CG \rightarrow\rightarrow H \}$

Does this relation in 4NF?

NO.

<u>Explanation:</u>

- since $A \rightarrow\rightarrow B$, but A is not a superkey for R

Fourth Normal Form (4NF)

- *Example*:
  - $R = (\underline{A}, B, \underline{C}, \underline{G}, H, I)$

    $F = \{ A \twoheadrightarrow B ;\ B \twoheadrightarrow HI ;\ CG \twoheadrightarrow H \}$

<u>Decompose to 4NF:</u>

Fourth Normal Form (4NF)

- *Example*:
  - $R = (\underline{A}, B, \underline{C}, \underline{G}, H, I)$

    $F = \{ A \rightarrow\rightarrow B ;\ B \rightarrow\rightarrow HI ;\ CG \rightarrow\rightarrow H \}$

Decompose to 4NF:

- Since $A \rightarrow\rightarrow B$, but A is not a superkey for R
- So, relation R is decomposed to,

- ***R1*** $= (\underline{A}, B)$

  $F = \{A \rightarrow\rightarrow B\}$

- ***R2*** $= (\underline{A}, \underline{C}, \underline{G}, H, I)$

  $F = \{CG \rightarrow\rightarrow H;\ A \rightarrow\rightarrow B;\ B \rightarrow\rightarrow HI \}$

It is in 4NF

It is not in 4NF as $CG \rightarrow\rightarrow H$
but CG is not a superkey of $R2$

Fourth Normal Form (4NF)

- *Example*:
  - $R = (\underline{A}, B, \underline{C}, \underline{G}, H, I)$

    $F = \{ A \rightarrow\rightarrow B ;\ B \rightarrow\rightarrow HI ;\ CG \rightarrow\rightarrow H \}$

<u>Decompose to 4NF:</u>

- **R1** = $(\underline{A}, B)$

  $F = \{A \rightarrow\rightarrow B\}$

- **R2** = $(\underline{A}, \underline{C}, \underline{G}, H, I)$

  $F = \{CG \rightarrow\rightarrow H;\ A \rightarrow\rightarrow B;\ B \rightarrow\rightarrow HI \}$

Fourth Normal Form (4NF)

- *Example*:
  - *R* =(<u>*A*</u>, *B*, <u>*C*</u>, <u>*G*</u>, *H*, *I*)

    F = { $A \rightarrow\rightarrow B$; $B \rightarrow\rightarrow HI$; $CG \rightarrow\rightarrow H$ }

<u>**Decompose to 4NF:**</u>

- ***R1*** = (<u>*A*</u>, *B*)

  F = { $A \rightarrow\rightarrow B$ }

- ***R2*** = (<u>*A*</u>, <u>*C*</u>, <u>*G*</u>, *H*, *I*)

  F = { $CG \rightarrow\rightarrow H$; $A \rightarrow\rightarrow B$; $B \rightarrow\rightarrow HI$ }

- ***R21*** = (<u>*C*</u>, <u>*G*</u>, *H*)

  F = { $CG \rightarrow\rightarrow H$ }

- ***R22*** = (<u>*A*</u>, <u>*C*</u>, <u>*G*</u>, *I*)

  F = { $A \rightarrow\rightarrow B$; $B \rightarrow\rightarrow I$ }

It is in 4NF

It is not in 4NF as $A \rightarrow\rightarrow I$
but *A* is not superkey of *R22.*

$B \rightarrow\rightarrow H$ is not preserved.

Fourth Normal Form (4NF)

- *Example*:
  - $R = (\underline{A}, B, \underline{C}, \underline{G}, H, I)$

    $F = \{ A \rightarrow\rightarrow B \;;\; B \rightarrow\rightarrow HI \;;\; CG \rightarrow\rightarrow H \}$

Decompose to 4NF:

- ***R1*** $= (\underline{A}, B)$

  $F = \{A \rightarrow\rightarrow B\}$

- ***R21*** $= (\underline{C}, \underline{G}, H)$

  $F = \{CG \rightarrow\rightarrow H\}$

- ***R22*** $= (\underline{A}, \underline{C}, \underline{G}, I)$

  $F = \{A \rightarrow\rightarrow B \;;\; B \rightarrow\rightarrow I \}$

Fourth Normal Form (4NF)

- *Example*:
  - $R = (\underline{A}, B, \underline{C}, \underline{G}, H, I)$

    $F = \{ A \rightarrow\rightarrow B;\ B \rightarrow\rightarrow HI;\ CG \rightarrow\rightarrow H \}$

<u>**Decompose to 4NF:**</u>

- ***R1*** $= (\underline{A}, B)$

  $F = \{A \rightarrow\rightarrow B\}$

- ***R21*** $= (\underline{C}, \underline{G}, H)$

  $F = \{CG \rightarrow\rightarrow H\}$

- ***R22*** $= (\underline{A}, \underline{C}, \underline{G}, I)$

  $F = \{A \rightarrow\rightarrow B;\ B \rightarrow\rightarrow I\}$

- ***R221*** $= (\underline{A}, I)$

  $F = \{A \rightarrow\rightarrow B;\ B \rightarrow\rightarrow I\}$

  It is in 4NF

- ***R222*** $= (\underline{A}, \underline{C}, \underline{G})$

  $F = \{ \}$

  It is in 4NF

Fourth Normal Form (4NF)

- *Example*:
  - $R = (\underline{A}, B, \underline{C}, \underline{G}, H, I)$

    $F = \{ A \rightarrow\rightarrow B ;\ B \rightarrow\rightarrow HI ;\ CG \rightarrow\rightarrow H \}$

Decompose to 4NF:

- **R1** = $(\underline{A}, B)$

  $F = \{A \rightarrow\rightarrow B\}$

- **R21** = $(\underline{C}, \underline{G}, H)$

  $F = \{CG \rightarrow\rightarrow H\}$

- **R221** = $(\underline{A}, I)$

  $F = \{A \rightarrow\rightarrow B ;\ B \rightarrow\rightarrow I \}$

- **R222** = $(\underline{A}, \underline{C}, \underline{G})$

  $F = \{ \}$

Summary of Learning

1) Normal Forms
   - 1NF
   - 2NF
   - 3NF
   - BCNF
   - 4NF

2) Decomposition Using Functional Dependencies
   - 1NF, 2NF, 3NF, BCNF

4) Decomposition Using Multivalued Dependency
   - 4 NF

Thank You

Acknowledgement: Almost all the tables and diagrams are taken from the book: A. Silberschatz, H. F. Korth, S. Sudarshan. Database System Concepts. 7th Ed, McGraw-Hill, 2019