Schema Normalization

R&G Chapter 19

In Last Lecture

- **An FD *F* can be *implied by* a set of FDs.**
- **F^+ = *closure of F* is the set of all FDs that are implied by *F*. (includes "trivial dependencies")**
- **How can we compute this closure?**
  - Approach 1: Rules of inference: 3 inference rules applied repeatedly to compute the closure.
  - Approach 2: Attribute closure calculation

Today's Class...

- **Normal forms and normalization**

Normal Forms

- **Back to the problem of schema refinement…**
- **A fundamental question: is any refinement needed??!**
- **Answer: if a relation is in a *normal form* :**
  - If a relation is in a certain *normal form*, certain problems are avoided/minimized.
  - Normal forms can help decide whether decomposition (i.e., splitting tables) will help.

Normal Forms

- **Types: 1st, 2nd, 3rd, Boyce-Codd**
- **1st ⊃ 2nd ⊃ 3rd ⊃ Boyce-Codd ⊃ …**
  - The higher the normal form is, the more constraints are put on the database

1st Normal Form

- **First Normal Form (1NF):**
  - Equivalent to the definition of relational model.
  - A relational schema is in 1NF if and only if the domains of all attributes of R are ***atomic***
    - A domain is atomic if elements of the domain are considered to be indivisible units.

Example

Course	Student
CS442	Alan Betty Carol

- **NOT 1NF form**

To make it satisfy 1NF

Course	Student
CS442	Alan
CS442	Betty
CS442	Carol

2nd Normal Form

- **Second Normal Form (2NF)**
  - *Partial dependency*: An FD X ->Y is said to be a partial dependency if there exists an FD Z ->Y such that $Z \subset X$.
  - For example
    - Given two FDs AB->C and A->C, AB->C is a partial dependency

2nd Normal Form

- **Second Normal Form (2NF): A relation schema R is in 2NF if each attribute A in R satisfies <u>one of</u> the following criteria:**
  1. A is part of a candidate key; OR
  2. A is not partially dependent on a candidate key.
     - For example: the key is {B, C}, the FD is B->A. Then A is partially dependent on the key.
- **In other words, a relation schema R is in 2NF if**
  - It is in 1NF
  - Each **non-key** attribute is dependent on the whole primary key (i.e., not partially dependent on a subset of key).

2NF ?

Staff

ENO	Name	Dno	DeptName	ProjNo	ProjName
E001	Somchai	D01	Physic	P01	NMR
E001	Somchai	D01	Physic	P02	Laser
E002	Sompong	D01	Physic	P03	Medical Image processing
E003	Somchay	D02	Computer Science	P05	Voice ordering
E003	Somchay	D02	Computer Science	P04	Speech Coding
E004	SomSiri	D02	Computer Science	P04	Speech coding
E004	SomSiri	D02	Computer Science	P06	Speech Synthesis

KEY = ENO + ProjNo

Answer is No. Because ProjName is dependent on ProjNo, a part of the key

Check Violation of 2NF

- **Given relation R and its FD F, if there exists any FD X->A in F^+ s.t.**

  (1) X is a subset of keys of R, and

  (2) A is a non-key attribute (i.e., A does not appear in any key)

Then R violates 2NF!

2NF Example

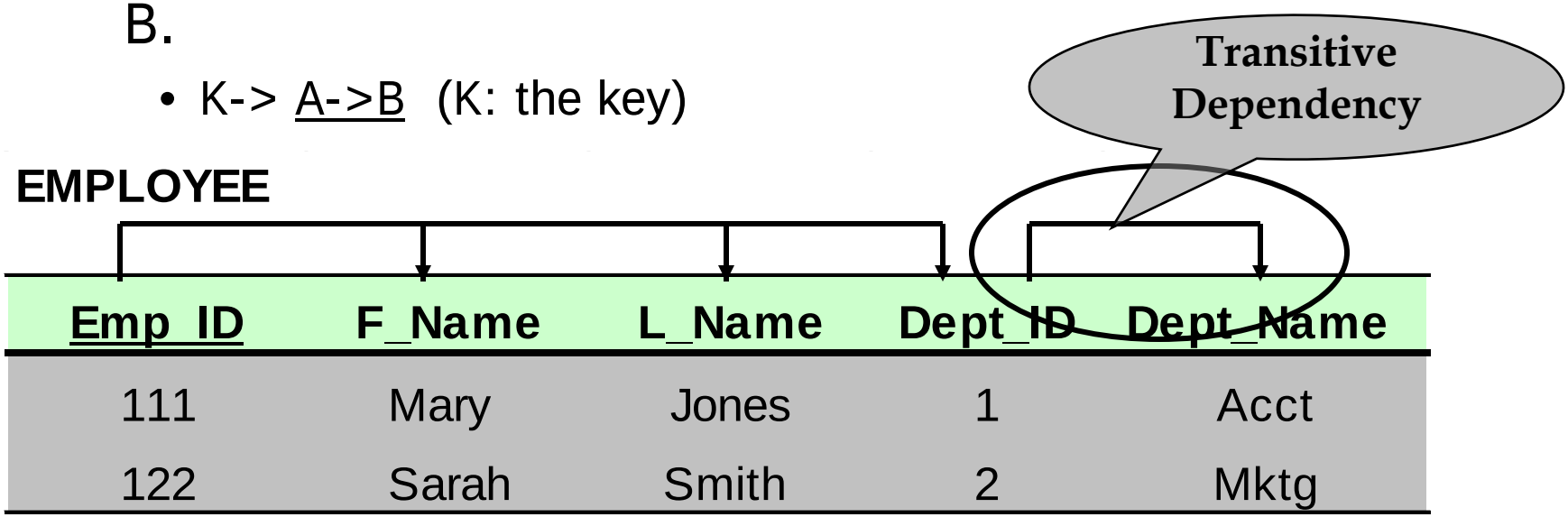
- **Relation R (A, B, C, D, E)**
- **F={ABD->C, BC->D, CD->E}**
- **Question: Is R a 2NF relation?**
- **Way of thinking:**
  - Step 1: find candidate keys of R
    - ABC^+=ABCDE, ABD^+=ABDCE.
    - Keys: ABC, ABD
  - Step 2: find non-/key attributes
    - key attributes: ABCD
    - Non key attribute: E

2NF Example

- **Relation R (A, B, C, D, E)**
- **F={ABD->C, BC->D, CD->E}**
- **Question: Is R a 2NF relation?**
- **Way of thinking:**
  - Step 3: check whether any subset of key functionally determines the non-key attribute
    - AB^+ = $ABAC^+$= AC
    - BC^+= BCDE (E is a non-key attribute), so BC->E!
- **Answer: R is not a 2NF relation.**

3rd Normal form

- **Transitive dependency**
  - When there exist two FDs: A->B, and B->C, then A-> C
  - There exist transitive dependency when a non-key attribute A determines another non-key attribute B.
    - K-> A->B (K: the key)



3rd Normal form

- **Relation R is said to be in 3rd Normal Form if:**
  - The relation R (table) is in 2NF, and
  - For each superkey K of R, all non-key attributes of R are directly dependent (i.e. non-transitively dependent) on K.

Check Violation of 3rd Normal Form

- **Two cases when X-> A violates 3NF**
  1. X is a subset of a key K.
     - This is a partial dependency, as K -> A.
  2. X is not a subset of the key while A is a non-key attribute.
     - This is a *transitive dependency*, as it has dependencies K ->X -> A.

3NF example

- **R = {A, B, C}**
- **F= {A->B, B->AC}**
- **Does R satisfy 3NF?**
- **Way of thinking**
  - Step 1: find candidate keys of R
    - A^+={ABC}, B^+={BCA}
    - Candidate key: A, B
  - Step 2: find non-/key attributes
    - Key attributes: A, B
    - Non-key attribute: C

3NF example (Cont.)

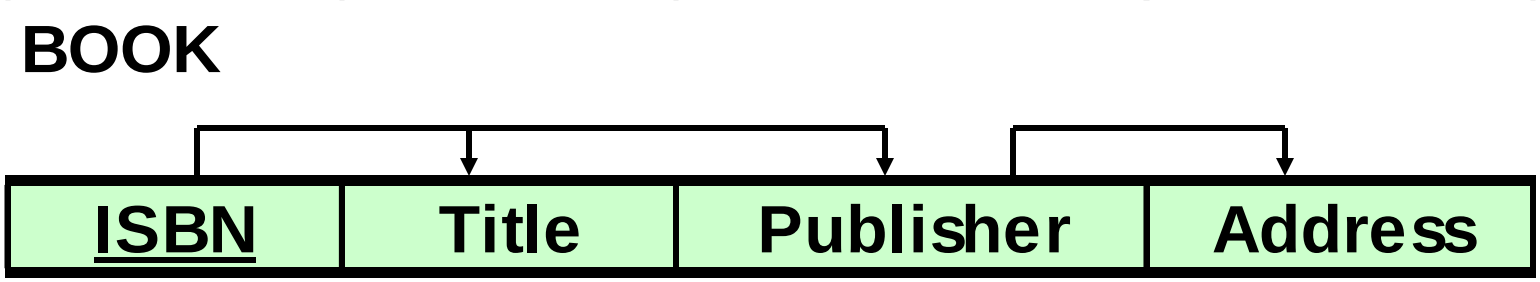
- **R = {A, B, C}**
- **F= {A->B, B->AC}**
- **Does R satisfy 3NF?**
- **Way of thinking**
  - Step 3: check 2NF violation
    - Each candidate key only consists of one attribute. There should be no partial dependency
  - Step 4: check 3NF violation
    - Is there any FD x->y s.t. x is not key?
      - Using A as key: B->AC violates
      - Using B as key: A->B violates
    - R violates 3NF!

Normal Forms: Review

- **Unnormalized – There are multivalued attributes or repeating groups**
- **1 NF – No multivalued attributes or repeating groups.**
- **2 NF – 1 NF + no partial dependencies**
- **3 NF – 2 NF + no transitive dependencies**

Example 1: Determine NF (1NF?)

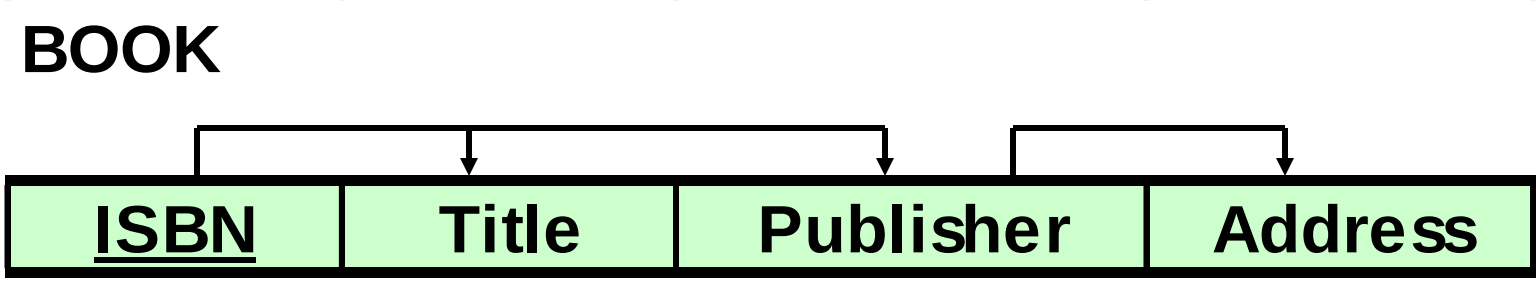
- **ISBN → Title**
- **ISBN → Publisher**
- **Publisher → Address**



The domains of all attributes of R are atomic, **therefore, the relation is at least in 1 NF**

Example 1: Determine NF (2NF?)

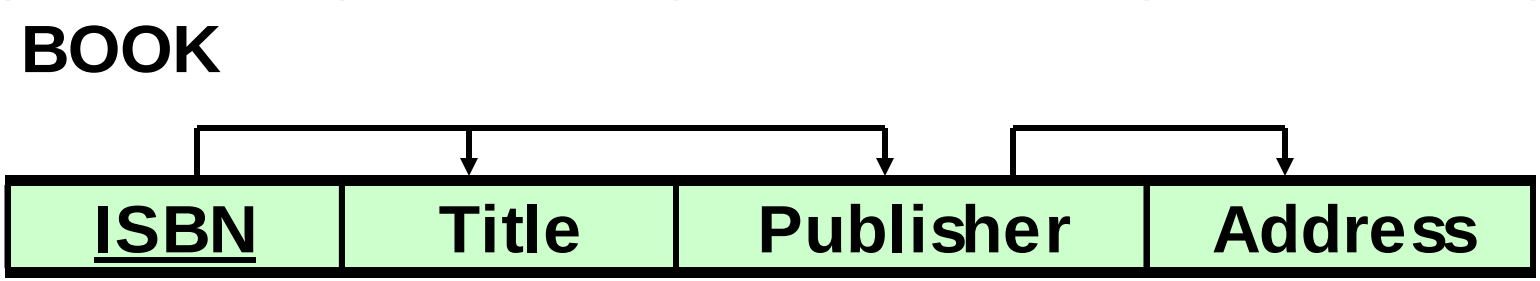
- **ISBN → Title**
- **ISBN → Publisher**
- **Publisher → Address**



There is no COMPOSITE primary key, therefore there can't be partial dependencies. Therefore, the relation is at least in 2NF

Example 1: Determine NF (3NF?)

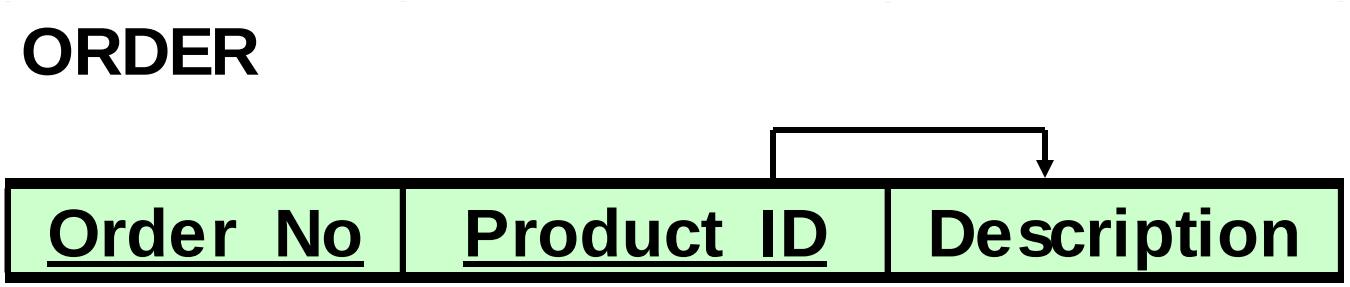
- **ISBN → Title**
- **ISBN → Publisher**
- **Publisher → Address**



It contains transitive dependency (ISBN-> Publisher, Publisher->Address). So the relation is not 3NF.

Example 2: Determine NF (1NF?)

- **Product_ID → Description**



All attributes are atomic; therefore, the relation is at least in 1 NF

Example 2: Determine NF (2NF?)

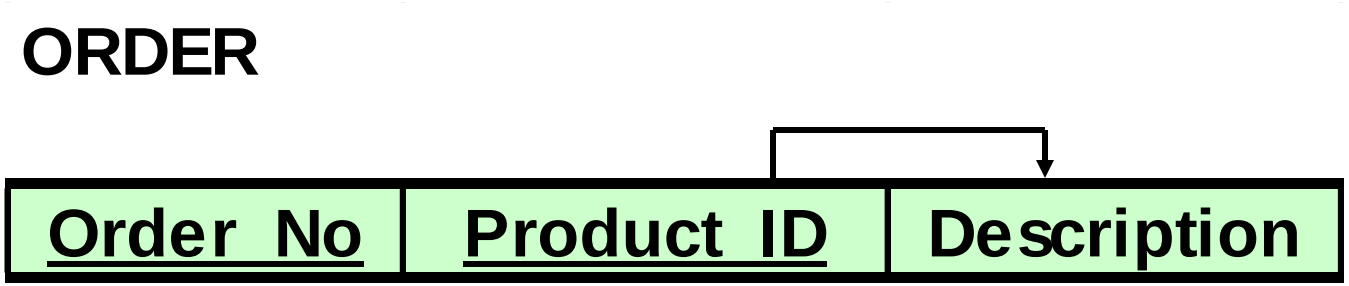
- **Product_ID → Description**

ORDER

Product_ID, which is a part of key, determines Description; hence, there is a partial dependency. Therefore, the relation is not 2NF

Example 2: Determine NF (3NF?)

- **Product_ID → Description**



No need to check 3NF, since it violates 2NF already!

Example 2: Determine NF

- **Product_ID → Description**

ORDER

We know that the relation is at least in 1NF, and it is not in 2 NF. Therefore, we conclude that the relation is in 1 NF.