Normal Forms

R&G Chapter 19

Normal Forms

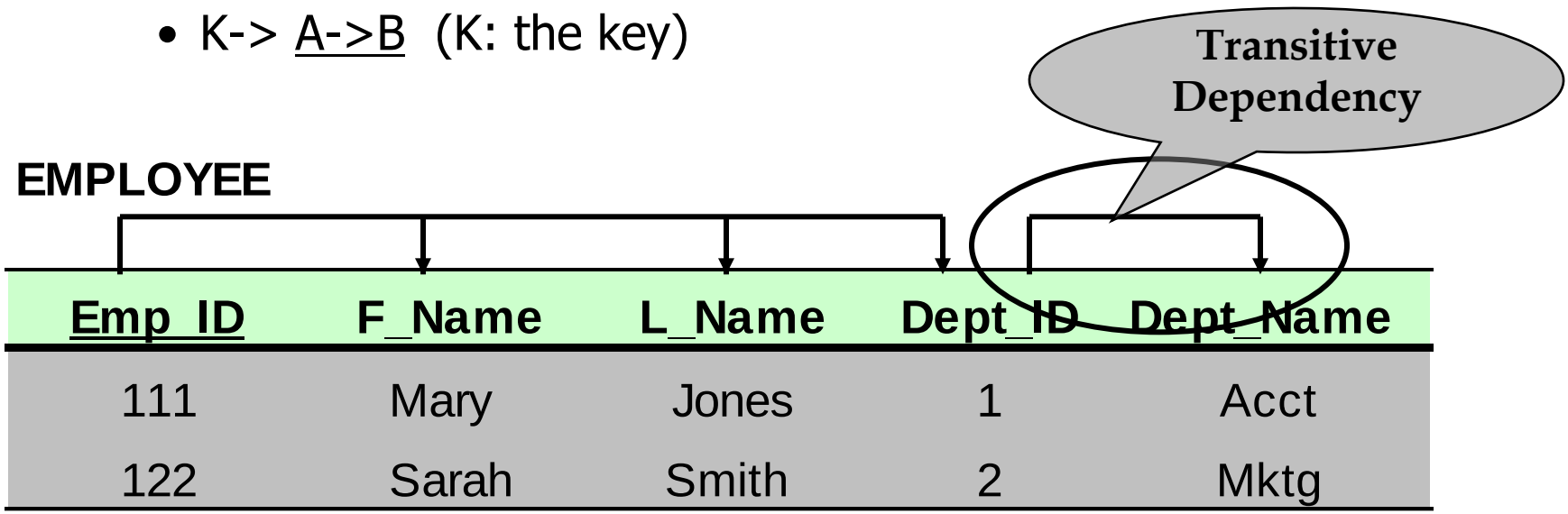
- Normal forms can help decide whether decomposition (i.e., splitting tables) will be needed to eliminate data redundancy.
  - If a relation is in a certain *normal form*, certain problems are avoided/minimized.
- Types: 1st, 2nd, 3rd, Boyce-Codd (3.5NF)
  - The higher the normal form is, the stricter constraints are put on the database

Normal Forms (NF)

- 1st NF (relational model)
- 2nd NF (free of partial dependency)
- **3rd NF**
- **Boyce-Codd (3.5NF)**

3rd Normal Form

- **Transitive dependency**
  - 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 both conditions are met:**
  - ***No partial dependency*** (i.e., the relation R is in 2NF)
  - ***No transitive dependency***: All non-key attributes of R are directly determined by a superkey.
    - In other word, for each given FD F: X->Y, it satisfies ONE of the following two conditions:
      - X is a superkey of R,
      - Y is a subset of some key (can be a single attribute)

Check Violation of 3rd Normal Form

- **There are 2 cases when a FD: X-> Y violates 3NF**

  Case 1: X is a subset of a key K.

  - This is a partial dependency, as K -> Y too.

  Case 2: X is not a subset of a key and Y is a non-key attribute.

  - This is a *transitive dependency*, as it has dependencies K ->X -> Y.

3NF example (1/2)

- **R = {A, B, C}**
- **F= {A->B, B->AC}**
- **Does R satisfy 3NF?**

3NF example (1/2)

- **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 (2/2)

(Continue with the previous slide)

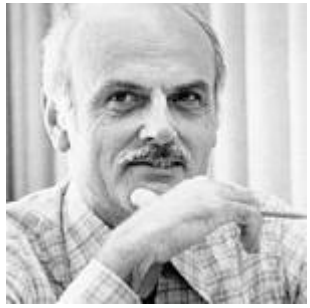
- Step 3: check 2NF violation
  - Each candidate key consists of only one attribute. There is no partial dependency
- Step 4: check 3NF violation
  - Is it true that for every single FD X->Y, either X is a key for R or Y is a subset of some key?
    - A->B: A is a key
    - B->AC: B is a key

- **Answer: R satisfies 3NF**

Shortcut Rules

- For any relation R
  - **Rule #1:** If all attributes of R are part of a key, R must be 2NF & 3NF (WHY?)
  - **Rule #2:** If R only contains singleton keys (i.e., each key only has one attribute), R must be at least 2NF

Boyce-Codd Normal Form (BCNF)



- **Also called 3.5NF.**
- **Reln R with FDs *F* is in BCNF if, for all X -> A in F^+, it satisfies one of the conditions:**
  - X is a superset of A (called a *trivial* FD), OR
  - X is a superkey for R.
- **In other words: "All non-trivial FDs must be FDs of key constraints"**

3NF VS BCNF

- **Given Reln R and its FDs F, for each FD X -> A in F^+,**
  - 3NF requires that ONE of the following two condition is met
    - (a) X is a key for R, **OR**
    - (b) A is part of some key (A can be a single attribute)
  - BCNF requires:
    - X is a superkey.

3NF VS. BCNF: Example I

- **R(ABC)**
- **Key of R: (A,B)**
- **F={C->B}**
- **Does R satisfy BCNF?**

- **Does R satisfy 3NF?**

3NF VS. BCNF: Example I

- **R(ABC)**
- **Key of R: (A,B)**
- **F={C->B}**
- **Does R satisfy BCNF?**
  - No, because in C->B, C is not a key
- **Does R satisfy 3NF?**
  - Yes, because B is a part of the key

3NF VS. BCNF: Example II

- **R(ABC)**
- **F={AB->C, C->A}**
- **Which normal form does R satisfy?**

Example of 3NF VS. BCNF

- R(ABC)
- F={AB->C, C->A}
- What normal form does R satisfy?
  - Step 1: Find the key of R:
    - AB, BC.
  - Step 2: Does it satisfy 2NF?
    - All attributes are part of some key, so it satisfies 2NF and 3NF (shortcut rule #1 on slide #10).
  - Step 3: Does it satisfy 3NF?
    - See above for 2NF
  - Step 4: Does it satisfy BCNF?
    - In C->A, C is not a key. So it does NOT satisfy BCNF.

Normal Form Summary

- **Types: 1st, 2nd, 3rd, Boyce-Codd**

Normal Form	Constraint
1NF	Atomic value
2NF	No partial dependency (i.e., there does not exist an FD X->A such that X is a subset of key and A is a non-key attr.)
3NF	No partial dependency & No transitive dependency (i.e., for each FD X->A, either X is a key or A is a subset of some key)
Boyce-Codd (3.5 NF)	All non-trivial FDs are key FDs (i.e., for each X->A, X is a superkey)