Normal Forms

SWEN304/SWEN435

Lecturer: Dr Hui Ma

Engineering and Computer Science

TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

Normalization

- Normalization is used to design a set of relation schemas that is optimal from the point of view of database updating
- Normalization starts from a universal relation schema
- There are six normal forms, of which only three are based on functional dependencies
- Normal forms define to which extent we should normalize
- The Synthesis algorithm and the Decomposition algorithm represent the formal normalization methods
- *Readings from the textbook:*
  - *Chapter 15* : *15.1-15.5,*
  - *Chapter 16* : *16.1 -16.3*

Normal Forms

- Normalization is a procedure that transforms a universal relation schema (U, F) into a set of relation schemas

$$S = \{N_i(R_i, K_i) \mid i = 1,\ldots, n\}$$

- The goal of the normalization is to avoid update anomalies by achieving a specified normal form
- There are six (vertical) normal forms defined in the theory of the relational data model
- These are: first, second, third, Boyce–Codd, fourth, and fifth normal form
- The second, third and Boyce–Codd normal form are based on functional dependencies

First Normal Form

- A relation schema is in **first normal form (1NF)** if the domain of its each attribute has only atomic values
  - No relation schema attribute is allowed to be composite or multi-valued
- Example:
  - *Student* (*StID, StName,* {*CourId, CoName, Grade*}) (*¬1NF*)
- Very often, the term "normalized relation" means "at least in the first normal form"
- From now on, if not otherwise noted, we shall consider only relation schemas that are at least in the first normal form

Second Normal Form

- A relation schema *R* is in **second normal form (2NF)** if no non-prime attribute in *R* is **partially** functionally dependent on any relation schema *R* key
- Example:

  $Grades$ ({$StID$, $StName$, $CourId$, $Grade$},

  {$StID \to StName$, $StID + CourId \to Grade$})

  $K(Grades) = StID + CourId$

  - is not in 2NF, but in 1NF, since
    - *Grade*, *StName* are non-prime attributes:
    - *Grade* is not partially (is fully) depended on the key
    - but *StName* is partially depended on the key
- Recall: non-prime attribute is an attribute that does not belong to any of the keys
- Second normal form relations still exhibits update anomalies

Third Normal Form

- A relation schema $N(R, F)$ with a set of keys $K(N)$ is in **third normal form** (3NF) if for each non-trivial functional dependency $X \to A$ holds in F, **either** X is a superkey of N, **or** A is a prime attribute of N
- X is a superkey of N: X is a superset of a key of N
- Formally 3NF can be defined by:

$$(\forall f : X \to A \in F)(A \in X \vee X \to R \in F^{+} \vee (\exists Y \in K(N))(A \in Y))$$

- Relation schemas being in the third but not in Boyce–Codd normal form still exhibit some update anomalies
- Recall: a prime attribute is a relation schema attribute that belongs to any of the keys

Another Definition of the Third Normal Form

- *According to Codd's original definition:*

A relation schema is in **third normal form (3NF)** if it is in 2NF, *and* no non-prime attribute is **transitively** functionally dependent on any relation schema key

- A functional dependency $X \to A$ in a relation schema N is a **transitive dependency** if there is a set Y that is neither a candidate key nor a subset of any key of N, and both $X \to Y$ and $Y \to A$ hold

- It can be proven that the two definitions given are equivalent

Third Normal Form – Examples (1)

- The relation schema

  Lecturer ({*LecId, LeName, CourId, CoName* }, {$LecId \rightarrow LeName$, $LecId \rightarrow CourId$, $LecId \rightarrow CoName$, $CourId \rightarrow CoName$ }),

  K (*Lecturer*) = *LecId*

  - It is in 2NF but not in 3NF,
  - since FD $CourId \rightarrow CoName$ holds in *F*, but neither *CourId* is a super key nor *CoName* is a prime attribute

Third Normal Form – Examples (2)

- The relation schema

  $Lecturer(\{LecId, LeName, CourId\}, \{LecId \rightarrow LeName, LecId \rightarrow CourId\}), K(Lecturer) = LecId$

  - Is in 3NF, at least
  - since all FDs in F have the LHS as a key

- The relation schema

  $N(\{A, B, C\}, \{A \rightarrow B, B \rightarrow A, B \rightarrow C\}), K = \{A, B\},$

  - Is it in 3NF?
  - Why?

Third Normal Form – Examples (3)

- Given $N(\{A, B, C\}, \{AB \rightarrow C, C \rightarrow B\})$, is N in 3NF?
  - We first need to determine minimal keys of N

Boyce-Codd Normal Form

- The Boyce-Codd normal form is the highest NF that is based on FDs
- The relation schema (R, F) is in **Boyce-Codd Normal Form (BCNF)**, if the left-hand side of each non-trivial functional dependency in F contains a relation schema key
- Formally

$$(\forall f: X \rightarrow A \in F)(A \in X \vee X \rightarrow R \in F^+)$$

Boyce-Codd Normal Form Examples (1)

- Employee={e_no, e_name, salary, child}
  with F = {e_no → e_name, e_no → salary}

  - Employee is not in BCNF wrt F
  - since
    - the FD e_no → e_name is not trivial, and
    - e_no is not a superkey for Employee wrt F:
      e_no^+ = {e_no, e_name, salary}

Boyce-Codd Normal Form Examples (2)

- INFO({e_no, e_name, salary}, {e_no → e_name, e_no → salary})
  - INFO is in BCNF wrt *F*
  - since
    - Both no trivial FDs e_no → e_name, e_no → salary have LHS as super key
      - e_no is a superkey for INFO wrt *F* :

      $e\_no^+ = \{e\_no, e\_name, salary\}$
- What about
    - INFO({e_no, e_name, salary}, {e_no → e_name, e_name → salary})?
    - $N(\{A, B, C\}, \{AB \to C, C \to B\})$,
  - Is it in BCNF wrt *F?*
  - Why?

Normal Form of a Set of Relation Schemas

- The normal form of a relation schema set

$$S = \{N_1(R_1, C_1), \dots, N_n(R_n, C_n)\}$$

  is determined by the normal form of the relation schema being in the lowest normal form
- Example:

$$S = \{N_1(\{A, B\}, \{A \to B\}),$$
$$N_2(\{B, C, D, E\}, \{BC \to D, C \to E\})\}$$

  - Due to N_2, S is in 1NF, even though N_1 is in BCNF
- Note: when considering normal forms, the set of constraints C is, often, considered as containing only functional dependencies

Normal Form Examples (1)

- let $R = CZS$ and $= \{Z \rightarrow C, CS \rightarrow Z\}$
  - determine minimal keys
  - Which normal form is it in?

- now take $R = ABCD$ and $= \{A \rightarrow B, B \rightarrow C, CD \rightarrow A, AC \rightarrow D\}$
  - determine minimal keys
  - Which normal form is it in?

Normal Form Examples (2)

- For $R = CZS$ and $F = \{Z \rightarrow C, CS \rightarrow Z\}$
  - We discover that the minimal keys are ZS and CS
  - Hence all attributes are prime and R is in 3NF

- For $R = ABCD$ and $F = \{A \rightarrow B, B \rightarrow C, CD \rightarrow A, AC \rightarrow D\}$
  - We discover that the minimal keys are A, BD and CD
  - Hence again all attributes are prime and R is in 3NF

- In both cases we did not have BCNF

Summary

- Of six normal forms defined in theory, only first four have significance in the practice
- Of these four only three are based on functional dependencies (2NF, 3NF, and BCNF)
- The first, second and (partly) third normal form suffer from update anomalies
- A set of BCNF relation schemas is (practically) free of update anomalies, and represents a possible goal of normalization
- The fact that a relation schema key functionally defines all relation schema attributes is crucial for understanding normal forms