Database Management System – 35 Database design – Minimal Set of Functional Dependencies and Keys

Ajay James
Asst. Prof in CSE
Government Engineering College Thrissur

Outline

- Minimal Set of Functional Dependencies
- Algorithm to find minimal cover
- Example
- Algorithm to find key

Minimal Sets of Functional Dependencies

- We applied inference rules to expand on a set F of FDs to arrive at F^+ (closure)
- Now think in the opposite direction to see if we could shrink or reduce the set F to its minimal form
  - so that the minimal set is still equivalent to the original set F

Minimal Sets of Functional Dependencies

- Informally, a **minimal cover** of a set of functional dependencies **E** is a set of functional dependencies F that satisfies the property that every dependency in **E is in the closure F^+ of F**
- F must have no redundancies in it
- Dependencies in F are in a standard form

Extraneous attribute - definition

- An attribute in a functional dependency is considered an **extraneous attribute** if we can remove it without changing the closure of the set of dependencies
- Given F, the set of functional dependencies, and a functional dependency $X \rightarrow A$ in F,
  - attribute Y is extraneous in X if $Y \subset X$,
  - and F logically implies $(F - (X \rightarrow A) \cup \{ (X - Y) \rightarrow A \})$

Extraneous attribute example

- R(A, B, C)
- $F = \{ AB \rightarrow C, A \rightarrow C \}$
- $F^+ = \{ AB \rightarrow C, A \rightarrow C \}$
- B is extraneous
- ***A → C, which means A alone can determine C***, the use of B is unnecessary (redundant)

- Given F, the set of functional dependencies, and a functional dependency $X \rightarrow A$ in F,
  - attribute Y is extraneous in X if $Y \subset X$,
  - and F logically implies

  $(F - (X \rightarrow A) \cup \{ (X - Y) \rightarrow A \})$

How to find extraneous attribute? Case 1

- $\alpha \rightarrow \beta$. Assume that α and β are set of one or more attributes.
- ***Case 1 (LHS)***: To find if an attribute A in α is extraneous or not
- **Step 1**: Find $(\{\alpha\} - A)^+$ using the dependencies of F
- **Step 2**: If $(\{\alpha\} - A)^+$ contains all the attributes of β, then A is extraneous.

Extraneous attribute example case 1

- F = {P→Q, PQ→R}. Is Q extraneous in PQ→R?
- *Step 1: Find $(\{\alpha\} - A)^+$ using the dependencies of F*
  - α is PQ. So find $(PQ - Q)^+$, ie., P^+
  - $P^+ = \{PQR\}$
- *Step 2: If $(\{\alpha\} - A)^+$ contains all the attributes of β, then A is extraneous.*
  - $(PQ - Q)^+$ contains R. Hence, Q is extraneous in PQ→R.
  - **{P→Q, P→R}**

How to find extraneous attribute? Case 2

- $\alpha \rightarrow \beta$. Assume that α and β are set of one or more attributes
- ***Case 2 (RHS)***: To find if an attribute A in β is extraneous or not
- **Step 1**: Find α^+ using the dependencies in F' where F' = (F – {$\alpha \rightarrow \beta$}) U { $\alpha \rightarrow (\beta - A)$ }.
- **Step 2**: If α^+ contains A, then A is extraneous.

Extraneous attribute example case 2

- F = {P→QR, Q→R}. Is R extraneous in P→QR?
- **Step 1**: Find α^+ using the dependencies in F' where F' = (F – {$\alpha \rightarrow \beta$}) U { $\alpha \rightarrow (\beta - A)$ }
  - F' = ({P→QR, Q→R} – {P→QR}) U {P→(QR-R)} = ({Q→ R} U {P→Q})
  - F' = {Q→R, P→Q}
  - Here, α is P. So find $(P)^+$, using the F'
  - Closure of P is {PQR}
- **Step 2**: If α^+ contains A, then A is extraneous.
  - P+ contains R. Hence, R is extraneous in P→QR
  - **F = {P→Q, Q→R}.**

Minimal set of FDs conditions

1. Every dependency in F has a single attribute for its right-hand side. (canonical/standard form)
2. We cannot replace any dependency $X \rightarrow A$ in F with a dependency $Y \rightarrow A$, where Y is a proper subset of X, and still have a set of dependencies that is equivalent to F
3. We cannot remove any dependency from F and still have a set of dependencies that is equivalent to F.

Minimal cover algorithm

- **Input: A set of functional dependencies E.**

1. Set F:=E.
2. Replace each functional dependency $X \rightarrow \{A_1, A_2, ..., A_n\}$ in F by the n functional dependencies $X \rightarrow A_1$, $X \rightarrow A_2$, ..., $X \rightarrow A_n$
3. For each functional dependency $X \rightarrow A$ in F for each attribute B that is an element of X if $\{ \{F - \{X \rightarrow A\} \} \cup \{ (X - \{B\}) \rightarrow A\} \}$ is equivalent to F then replace $X \rightarrow A$ with $(X - \{B\}) \rightarrow A$ in F. **(* The above constitutes a removal of the extraneous attribute B from X *)**
4. For each remaining functional dependency $X \rightarrow A$ in F if $\{F - \{X \rightarrow A\} \}$ is equivalent to F, then remove $X \rightarrow A$ from F.

   (* The above constitutes a removal of the redundant dependency $X \rightarrow A$ from F *)

Example 1

- *E : {B → A, D → A, AB → D}*
- *Step 1 – canonical form*
- *Step 2 – extraneous attributes in LHS*
  - *Check for AB → D*
  - *Remove A, find B^+*
  - *B^+ = {BAD}, B^+ contains D, so A is extraneous*
  - *E : {B → A, D → A, B → D}*
- Step 3: Redundant FD
  - *B → D, D → A implies B → A (transitivity rule)*
- ***F={D → A, B → D} (minimal cover)***

Example 2

- G: {A → BCDE, CD → E}.
- **Step 1:** G: {A → B, A→ C, A→ D, A→ E, CD → E}
- **Step 2:** Redundant/extraneous attribute in LHS
  - CD → E, remove C
  - Find D^+ = {D}
  - CD → E, remove D
  - Find C^+ = {C}
  - So, C and D are not redundant attributes

Example 2 contd...

- G: {A → B, A→ C, A→ D, A→ E, CD → E}
- A→ C, A→ D implies A → CD
- A → CD and CD → E implies A→ E
- So A→ E is redundant
- **Minimal cover**
  - **F = {A → B, A→ C, A→ D, CD → E}**
  - **Or F= {A → BCD, CD → E}**

Algorithm to find key of a relation

- **Input: A universal relation R and a set of functional dependencies F on the attributes of R**

1. Set K := R;
2. For each attribute A in K

   {

   Compute (K -A)+ with respect to F;

   If $(K - A)^{+}$ contains all the attributes in R,

   then set K := K -{A};

   }

Reference

- Elmasri R. and S. Navathe, Database Systems: Models, Languages, Design and Application Programming, Pearson Education 6th edition and 7th edition

Thank you