Database Management System
Normalization
UNIT – II
Part-II

Closure of a Set of Functional Dependencies

For a set F of functional dependencies, we call the closure of F, noted F+, the set of all the functional dependencies that can be derived from F (by the application of Armstrong's axioms).

Intuitively, F+ is equivalent to F, but it contains some additional FDs that are only implicit in F.

Closure of a Set of Functional Dependencies

For a set X of attributes, we call the **closure** of X *(with respect to a set of functional dependencies F)*, noted X+, the maximum set of attributes such that X→X+ *(as a consequence of F)*

Example: Consider the relation scheme R(A,B,C,D) with functional dependencies {A}→{C} and {B}→{D}.

- {A}+ = {A,C}
- {B}+ = {B,D}
- {C}+={C}
- {D}+={D}
- {A,B}+ = {A,B,C,D}

{A,B} is a superkey because:

- It determines all attributes

Closure Property

Example : R = (A, B, C, G, H, I)

$F = \{ A \rightarrow B, \; A \rightarrow C, CG \rightarrow H, CG \rightarrow I, \; B \rightarrow H\}$

Find some members of F^+

Closure Property

- $R = (A, B, C, G, H, I)$
  $F = \{ A \rightarrow B,\ A \rightarrow C, CG \rightarrow H, CG \rightarrow I,\ B \rightarrow H\}$
- some members of F^+
  - $A \rightarrow H$ (by transitivity from $A \rightarrow B$ *and* $B \rightarrow H$)
  - $AG \rightarrow I$ ($A \rightarrow C$

    $AG \rightarrow CG$

    $CG \rightarrow I$ given

    $AG \rightarrow I$

    Augmentation rule
    $X \rightarrow Y;\ XZ \rightarrow YZ$

    Transitivity rule
    $X \rightarrow Y;\ XZ \rightarrow YZ$

    - by augmenting $A \rightarrow C$ with G, to get $AG \rightarrow CG$ and then transitivity with $CG \rightarrow I$
  - $CG \rightarrow HI$
    - By Secondary rule Union.

Closure of Functional Dependencies

- We can further simplify manual computation of F^+ by using the following additional rules.
  - If $\alpha \to \beta$ holds *a*nd $\alpha \to \gamma$ holds, then $\alpha \to \beta\gamma$ holds ***(union)***
  - If $\alpha \to \beta\gamma$ holds, then $\alpha \to \beta$ holds and $\alpha \to \gamma$ holds ***(decomposition)***
  - If $\alpha \to \beta$ holds *a*nd $\gamma\beta \to \delta$ holds, then $\alpha\gamma \to \delta$ holds ***(pseudotransitivity)***

  The above rules can be inferred from Armstrong's axioms.

Exercise: The Closure of Attributes

- R = (A, B, C, D, E) and

  F = {A→BC, CD→E, B→D, E→A}.

  Compute A^+ and B^+.

Now Compute B^+

R = (A, B, C, D, E) and

$F = \{A \to BC, CD \to E, B \to D, E \to A\}$

Solution:

$B^+ = B$ because $B \to B$

$= BD$ because $B \to D$

Q1. **$R = (A, B, C, D, E)$ and $F = \{A \rightarrow BC, CD \rightarrow E, B \rightarrow D, E \rightarrow A\}$. List all candidate keys of R.**

Q1. R = (A, B, C, D, E) and

F = {A→BC, CD→E, B→D, E→A,A →E}.

- **List all candidate keys of R.**

Sol. (1) Since A→B and B→D, then A→D. (decomposition, transitivity)

Since CD →A and CD→E, then A→E E →A.

(decomposition, union and transitivity)

A →A. (reflexivity)

Therefore A → ABCDE.

So A is a candidate key of R.

(2) Since E→A, then E→ABCDE. (transitivity)

So E is also a candidate key of R.

(3) Since CD→E, then CD→ABCDE. (transitivity)

So CD is also a candidate key of R.

(4) Since B→D and BC→CD, then BC →ABCDE. (augmentation, transitivity)

So BC is also a candidate key of R.

Let us apply our rules to the example of schema R = (A, B, C, G, H, I) and the set F of functional dependencies {A → B, A → C, CG → H, CG → I, B → H}.

We list several members of F+ here:

• A → H. Since A → B and B → H hold, we apply the transitivity rule. Observe

• CG → HI . Since CG → H and CG → I , the union rule implies that CG → HI .

• AG → I. Since A → C and CG → I, the pseudotransitivity rule implies that AG → I holds.

Another way of finding that AG → I holds is as follows. We use the augmentation rule on A → C to infer AG → CG. App

Compute the closure of the following set *F of functional dependencies* for relation schema $R = (A, B, C, D, E)$.

$$A \rightarrow BC$$

$$CD \rightarrow E$$

$$B \rightarrow D$$

$$E \rightarrow A$$

List the candidate keys for R.

Starting with $A \rightarrow BC$, we can conclude:
$A \rightarrow B$ and $A \rightarrow C$.

Since $A \rightarrow B$ and $B \rightarrow D$, $A \rightarrow D$ (decomposition, transitive)

Since $A \rightarrow CD$ and $CD \rightarrow E$, $A \rightarrow E$ (union, decomposition, transitive)

Since $A \rightarrow A$, we have (reflexive)

$A \rightarrow ABCDE$ from the above steps (union)

Since $E \rightarrow A$, $E \rightarrow ABCDE$ (transitive)

Since $CD \rightarrow E$, $CD \rightarrow ABCDE$ (transitive)
Since $B \rightarrow D$ and $BC \rightarrow CD$, $BC \rightarrow ABCDE$ (augmentative, transitive)
Also, $C \rightarrow C$, $D \rightarrow D$, $BD \rightarrow D$, etc.

Therefore, any functional dependency with *A*, *E*, *BC*, *or CD on the left hand* side of the arrow is in *F+*, *no matter which other attributes appear in the FD*

Algorithm for finding Closure of Attribute Sets

An Algorithm to compute α^+, the closure of α under F

result := **α;**

repeat

for each functional dependency $\beta \rightarrow \gamma$ in F do

begin

if $\beta \subseteq$ *result* ***then*** *result* := *result* U γ ;

end

until (*result* does not change)

Lossy/Lossless Join

Q. R(A,B,C,D,E,) Find the given relation is lossy or lossless join

R1(A,B,C)

R2(A,D,E)

A→ BC

Method:

1.a Fill in the form of ai

b. Rest value fill in the form of bi

2. Check for all given FD

change bi to ai if required

3. If any row under all column filled by a then decomposition is lossyless join. Otherwise not.

4.Stop

Step-1

	A	B	C	D	E
R1	a1	a2	a3	b4	b5
R2	a1	b2	b3	a4	a5

Step-2 A→BC given

	A	B	C	D	E
R1	a1	a2	a3	b4	b5
R2	a1	~~b2~~ **a2**	~~b3~~ **a3**	a4	a5