SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

Re-accredited by NAAC with A+ grade, Accredited by NBA(CSE, IT, ECE, EEE & Mechanical)
Approvedy by AICTE, New Delhi, Recognized by UGC, Affiliated to Anna University, Chennai

Department of MCA

DBMS Dependency Preserving Decomposition

Course Name : 23CAT603 - DATA BASE MANAGEMENT SYSTEM

Class : I Year / I Semester

Unit III – Dependency Preserving Decomposition

Dependency Preserving Decomposition

Dependency Preservation: A Decomposition D = { R1, R2, R3…Rn } of R is dependency preserving wrt a set F of Functional dependency if

(F1 ? F2 ? … ? Fm)+ = F+.

Consider a relation R R ---> F{...with some functional dependency(FD)....}

R is decomposed or divided into R1 with FD { f1 } and R2 with { f2 }, then there can be three cases:

f1 U f2 = F -----> Decomposition is dependency preserving.

f1 U f2 is a subset of F -----> Not Dependency preserving.

f1 U f2 is a super set of F -----> This case is not possible.

Dependency Preserving Decomposition

Problem:

Let a relation R (A, B, C, D) and functional dependency {AB –> C, C –> D, D –> A}. Relation R is decomposed into R1(A, B, C) and R2(C, D). Check whether decomposition is dependency preserving or not.

Solution:

R1(A, B, C) and R2(C, D)
Let us find closure of F1 and F2 To find closure of F1,
consider all combination of ABC. i.e.,
find closure of A, B, C, AB, BC and AC Note ABC is not considered as it is always ABC
closure(A) = { A } // Trivial
closure(B) = { B } // Trivial
closure(C) = {C, A, D} but D can't be in closure as D is not present R1.
= {C, A}
C--> A // Removing C from right side as it is trivial attribute

Dependency Preserving Decomposition

closure(AB) = {A, B, C, D} = {A, B, C}
AB --> C // Removing AB from right side as these are trivial attributes
closure(BC) = {B, C, D, A} = {A, B, C}
BC --> A // Removing BC from right side as these are trivial attributes
closure(AC) = {A, C, D}
NULL SET
F1 {C--> A, AB --> C, BC --> A}.
Similarly F2 { C--> D }
In the original Relation Dependency
{ AB --> C , C --> D , D --> A}.
AB --> C is present in F1.
C --> D is present in F2.
D --> A is not preserved.
F1 U F2 is a subset of F. So **given decomposition is not dependency preserving.**

References

1. https://lnct.ac.in/wp-content/uploads/2020/03/Unit-4-Dependency-preservation-notes-.pdf
2. https://www.geeksforgeeks.org/data-base-dependency-preserving-decomposition/