RESUMEN NORMALIZACIÓN EN 3FN

Algoritmo de normalización en 3FN (algoritmo de síntesis)

Dados R y F, generar una descomposición de producto sin pérdida en varios esquemas Ri, que estará en BCNF si es posible conservar dependencias, o bien en 3FN en otro caso.

0. Partir de un recubrimiento canónico de F
1. Calcular claves candidato de R
2. Para cada dependencia funcional de F:
    a. Generar un esquema Ri para esa dependencia
    b. Calcular la restricción Fi de F para Ri
    c. Calcular claves candidato de Ri
3. Si hay alguna dependencia de F que se puede comprobar en más de un esquema Ri, eliminarla de donde sea posible (donde no afecte a la conservación de dependencias) (eliminar = quitar la dependencia del Fi y los *atributos de la derecha* de Ri)
4. Si ninguna clave candidato de R aparece en alguno de los esquemas Ri resultantes, añadir un esquema adicional para una clave candidato de R (vale cualquiera)
5. Indicar la forma normal de cada relación Ri (será BCNF o 3FN)
6. Comprobar que es una descomposición de producto sin pérdida (siempre lo debería ser)
7. Comprobar que se conservan las dependencias (siempre debería suceder)

3NF NORMALIZATION - SUMMARY

3NF Normalization Algorithm (synthesis algorithm)

Given R and F, generate a lossless join decomposition into several Ri schemas. Ri will be in BCNF if possible (preserving dependencies), and otherwise in 3NF .

0. Start with a minimal cover for F
1. Compute candidate keys for R
2. For each functional dependency in F:
    a. Generate an Ri schema for that dependency
    b. Compute the Fi constraint of F for Ri
    c. Compute candidate keys for Ri
3. If there is some dependency in F that can be checked in more than one schema Ri, delete where possible (where dependency preservation is not affected) (delete = remove the dependency from Fi, and delete the *right side attributes* from Ri).
4. If no candidate key for R appears in some Ri schema, add an additional schema for one candidate key (any one).
5. Mark the normal form for each schema (it will be BCNF of 3NF)
6. Check it is indeed a lossless join decomposition (it should be)
7. Check that depencies are preserved (they should be)

EJERCICIOS / EXERCISES

A) Encontrar una descomposición de producto sin pérdida (PSP), que conserve las dependencias (CD), y normalizada en 3FN o BCNF lo mejor posible (mínimo número de esquemas y de redundancia, etc.) del esquema de relaciones R1 = (A, B, C, D, E, F, G, H, I) y el conjunto de dependencias F:
Find a lossless join (LJ), dependency preserving (DP) decomposition, normalized as best as possible in 3NF or BCNF (minimal number of schemas and redundancy), using the schema R1 = (A, B, C, D, E, F, G, H, I), and the set of functional dependencies F:

F1 = {
    D E --> C F
    B C --> A G H
    F --> I
    A --> D E
}

Indicar la forma normal de cada relación y mostrar que la descomposición encontrada efectivamente es de producto sin pérdida y conserva las dependencias.
Mark the normal form of each relation. Show that the decomposition is in fact LJ and DP.

B) Lo mismo con / same with

R2 = (A, B, C, D, E, F, G, H)

F2 = {
    B C --> D
    D E --> G
    C D --> A E
    D --> B H
}