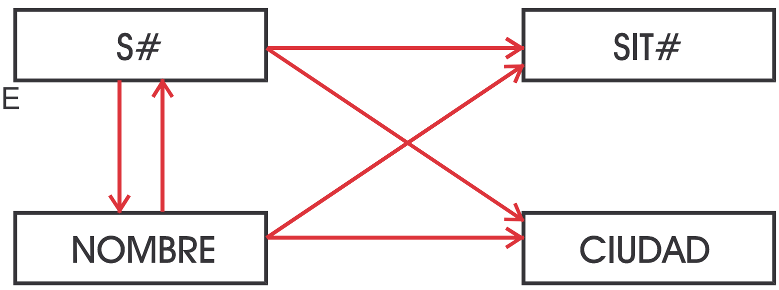
Normalización de un esquema relacional. Algoritmo para encontrar todas las Claves Candidatas

Sean R(X,Z) un esquema relacional, *F* un conjunto de dependencias funcionales sobre ese esquema y *D* el conjunto de determinantes *F* , donde X representa el conjunto de atributos de R que no son determinados por ninguna dependencia de *F* y Z son el resto de los atributos:

1. Si $X \neq \emptyset$ entonces usando *F* y Axiomas de Amstrong probar si $X \rightarrow Z$?
    1. Si $X \rightarrow Z$ entonces X es la única clave candidata. Fin del procedimiento
    2. Si $X \not\rightarrow Z$
        1. Todas las combinaciones de la forma (X,z_i) con $z_i \subseteq Z$ que cumplan $(X,z_i) \rightarrow Z$ y que cumplan el criterio de minimalidad serán claves candidatas. Fin del procedimiento
2. Si $X = \emptyset$ entonces, usando *F* y Axiomas de Amstrong, para cada determinante de *D* habrá que evaluarse si, sólo o en combinación con otros atributos de R, determinan de forma mínima al resto de los atributos de R, en cuyo caso, será clave candidata. Fin del procedimiento

Normalización de un esquema relacional. Ejemplos

- Ejemplo 1.
  S(S#,NOMBRE,SIT#,CIUDAD)
  - Suponemos que S# y NOMBRE son claves candidatas
  - No existe dependencia: S.CIUDAD->S.SIT#
- Solución:
  - Diagrama de dependencias funcionales



Está en FNBC porque las únicas flechas que salen, lo hacen de las claves candidatas

- Ejemplo 2. SSP(S#,NOMBRE,P#,QTY)
  - Claves candidatas (S#,P#) y (NOMBRE,P#)
  - S# ↔ Nombre
- Solución:
  - No está en FNBC porque tiene dos determinantes S# y NOMBRE que no son claves candidatas
  - Descomposiciones posibles:
    - Descomposición 1: SS(S#,NOMBRE) y SP(S#,P#,QTY)
    - Descomposición 2: SS(S#,NOMBRE) y SP(NOMBRE,P#,QTY)