



Trabajo Práctico 4

Algoritmos de Normalización

1) Se tiene un esquema de relación **Personas** con los siguientes datos: **DNI, Nombre, Dirección, Localidad, Código Postal, Nombre Hijo, Edad Hijo, Escuela donde vota, Dirección Escuela, Localidad Escuela, Código Postal Escuela**. Se conocen las siguientes dependencias funcionales:

Código Postal \rightarrow Localidad

Localidad \rightarrow Código Postal

Escuela donde vota, Localidad \rightarrow Dirección Escuela, Código Postal Escuela

DNI, Nombre Hijo \rightarrow Edad Hijo

1. Explicar detalladamente cuáles son las anomalías que presenta el esquema.
2. Hallar una clave.
3. Descomponer el esquema de tal forma que ya no se presenten las anomalías detectadas.
4. Indicar si el esquema obtenido en el punto anterior cumple con la propiedad lossless join y preservación de dependencias.

2) Implemente el algoritmo escrito en pseudocódigo, de modo que cuando reciba una relación R , un conjunto F de dependencias funcionales y un conjunto de atributos A , calcule y devuelva A^+ . Incluya en el informe la salida presentada por su programa en cada uno de los set de prueba.

Cierre de conjuntos de atributos α^+ :

```
resultado := alfa;  
while (cambios en resultado) do  
  for each dependencia funcional  $b \rightarrow c$  in  $F$  do  
    begin  
      if  $b \subseteq \text{resultado}$  then resultado := resultado  $\cup$   $c$ ;  
    end
```

3) Diseñe un algoritmo que dada una relación R y un conjunto de dependencias funcionales F devuelva el conjunto **de todas sus claves candidatas**, para esto puede utilizar el cierre de un conjunto de atributos que haya implementado en el punto anterior. **Importante:** no será evaluada la optimalidad de la solución propuesta (complejidad temporal, espacial, etc), pero sí que funcione correctamente en todos los casos.

Debe presentar tanto su algoritmo en pseudo-código junto con una breve explicación dentro del informe. Además incluya la salida presentada por su programa en cada uno de los sets de prueba.

Sets de prueba

Set 1

R1={A, B, C, D, E, F, G, H, I, J}
F1={AB->C, BD->EF, AD->GH, A->I, H->J}
A1={B, D}

Set 2

R2={A, B, C, D, E, F, G, H}
F2={A->BC, C->D, D->G, H->E, E->A, E->H}
A2={A, C}

Set 3

R3={A, B, C, D, E, F, G}
F3={A->G, A->F, B->E, C->D, E->A, D->B, GF->C}
A3={F, G}

Consideraciones generales

Tener en cuenta que:

- El informe debe contener la resolución del ejercicio 1, la salida resultante del algoritmo implementado para los sets de prueba para los ejercicios 2 y 3, y el algoritmo diseñado en pseudo código con una breve explicación para el ejercicio 3.
- Puede utilizarse cualquier lenguaje para implementar los algoritmos, y se incluirá los archivos de código fuente en la entrega. Estos deben estar bien tabulados, las variables y funciones usadas tienen que tener nombre significativos y ser legibles!
- Se recomienda elegir un lenguaje de programación que incluya la estructura de datos Set o en su defecto incorporarla con alguna librería o hacer una implementación propia. No debe mezclar en un mismo método lógica de los algoritmos pedidos y lógica que evite repetidos o similar.
- Los sets de prueba pueden estar "hard-coded" en el código fuente o ser leídos del teclado o un archivo.

El trabajo se realiza en grupos de a lo sumo tres alumnos.

Documentación a entregar:

1. Un pequeño informe donde consten las aclaraciones que considere necesarias, el ejercicio 1 y la explicación del algoritmo propuesto en el ejercicio 3. Además debe incluirse la salida del programa para los sets de datos propuestos.
2. Los archivos de código fuente de los algoritmos implementados.

Forma de Entrega: Entregar el informe en papel y enviar un archivo comprimido con el informe y demás archivos por Email a: diana@fceia.unr.edu.ar