**Ejemplos algoritmo para descomposición 3NF de una relación R**

**Ejemplo 1**

Para la relación **MATRÍCULA** (cédula, nom\_afición, valor), CP = {cédula} y con S:

{

cédula 🡪 nom\_afición

nom\_afición 🡪 valor

cédula 🡪 valor

}

**Aplicación del algoritmo**

1. Hallar **S-**. El cierre mínimo de S es el mismo del Ejemplo 1 del documento de ejemplos de cierre mínimo. O sea:

{

cédula 🡪 nom\_afición

nom\_afición 🡪 valor

}

2.

s1 = { cédula 🡪 nom\_afición}

s2 = { nom\_afición 🡪 valor}

3.

R1 = (cédula, nom\_afición) con CP = {cédula}

R2 = (nom\_afición, valor) con CP = {nom\_afición}

4. La CP de **MATRÍCULA** (o sea, {cédula}) ya está en R1.

5. No hay esquemas redundantes.

**Resultado**:

R1 = (cédula, nom\_afición) con CP = {cédula}

R2 = (nom\_afición, valor) con CP = {nom\_afición}.

**Ejemplo 2a**

Para la relación **ENVÍO** (NIT, producto, cant, sede\_ppal, IVA), CP = { NIT, producto } y con S:

{

NIT producto 🡪 cant

NIT 🡪 sede\_ppal

producto 🡪 IVA

}

1. Hallar **S-**. El cierre mínimo de S es el mismo S:

{

NIT producto 🡪 cant

NIT 🡪 sede\_ppal

producto 🡪 IVA

}

2.

s1 = { NIT producto 🡪 cant }

s2 = { NIT 🡪 sede\_ppal }

s3 = { producto 🡪 IVA }

3.

R1 = (NIT, producto, cant) con CP = {NIT, producto}

R2 = (NIT, sede\_ppal) con CP = {NIT}

R3 = { Producto, IVA) con CP = {producto}

4. La CP de **ENVÍO** (o sea, {NIT, producto}) ya está en R1.

5. No hay esquemas redundantes.

**Resultado**:

R1 = (NIT, producto, cant) con CP = {NIT, producto}

R2 = (NIT, sede\_ppal) con CP = {NIT}

R3 = { producto, IVA) con CP = {producto}

**Ejemplo 2b**

**Nota**: A diferencia del ejemplo **2a**, aquí no hay atributo cant.

Para la relación **ENVÍO** (NIT, producto, sede\_ppal, IVA), CP = { NIT, producto } y con S:

{

NIT 🡪 sede\_ppal

producto 🡪 IVA

}

1. Hallar **S-**. El cierre mínimo de S es el mismo S:

{

NIT 🡪 sede\_ppal

producto 🡪 IVA

}

2.

S1 = { NIT 🡪 sede\_ppal }

S2 = { producto 🡪 IVA }

3.

R1 = (NIT, sede\_ppal) con CP = {NIT}

R2 = { producto, IVA) con CP = {producto}

4. La CP de **ENVÍO** (o sea, {NIT, producto}) no está en R1 ni en R2, entonces se crea R0 así:

R0 = (NIT, producto)

5. No hay esquemas redundantes.

**Resultado**:

R0 = (NIT, producto) con CP = {NIT, producto}

R1 = (NIT, sede\_ppal) con CP = {NIT}

R2 = { producto, IVA) con CP = {producto}

**Ejemplo 3**

Para la relación **MATRÍCULA\_SEMESTRE** (carné, ced\_est, materia, nota\_def)con S:

{

carné 🡪 ced\_est

ced\_est 🡪 carné

carné materia 🡪 nota\_def

ced\_est materia 🡪 nota\_def

}

**Nota:** Esta relación tiene dos claves candidatas ( {carné, materia} y {ced\_est, materia} ); por lo tanto, la definición de 3NF vista NO es aplicable. Sin embargo, veamos que pasa al aplicar el algoritmo.

**Aplicación del algoritmo**

1. Hallar **S-**. El cierre mínimo de S es el mismo del Ejemplo 3 del documento de ejemplos de cierre mínimo. Allí se vio que S tenía, de hecho, dos cierres mínimos. Tomemos uno de ellos (al final se muestra el resultado si se hubiera tomado el otro cierre):

{

carné 🡪 ced\_est

ced\_est 🡪 carné

ced\_est materia 🡪 nota\_def

}

2.

s1 = { carné, ced\_est }

s2 = { ced\_est, carné }

s3 = { ced\_est, materia, nota\_def }

3.

R1 = (carné, ced\_est) con CP = {carné}

R2 = (ced\_est, carné) con CP = {ced\_est}

R3 = (ced\_est, materia, nota\_def) con CP = { ced\_est, materia }

4. Una clave de **MATRÍCULA\_SEMESTRE** (o sea, { ced\_est, materia }) ya está en R3.

**Nota**: Como el algoritmo es para descomposición 3NF (R debe tener **una sola** clave candidata), como acá tiene **dos** claves candidatas entonces el paso 4 en el algoritmo queda así:

Si ningún esquema Ri incluye una clave (candidata) de R (sin embargo, recuérdese que este algoritmo está concebido para relaciones con una sola clave candidata).

5. Los esquemas R1 y R2 son equivalentes. Uno de los dos sobra.

**Resultado**:

R1 = (carné, ced\_est) con CP = {carné}

Nota: en R1, ced\_est es clave (alternativa).

R3 = (ced\_est, materia, nota\_def) con CP = { ced\_est, materia }

**Nota: Si se hubiera tomado como cierre mínimo el otro, el resultado hubiese sido:**

R1 = (carné, ced\_est) con CP = {carné}

Nota: en R1, ced\_est es clave (alternativa).

R3 = (carné, materia, nota\_def) con CP = { carné, materia }

En este ejemplo se obtuvo una descomposición 3NF y que de hecho es BCNF. Sin embargo, recordar que el algoritmo está concebido solamente para 3NF.