Normalización

BASES DE DATOS

Mercedes García

Normalización

- Diseño de una base de datos
  - Definición del diagrama ER que represente el problema.
  - Transformación del diagrama ER al modelo relacional.
  - Implementación en el SGBD elegido dando lugar a un conjunto de tablas mas un conjunto de restricciones
- Una tarea más…**normalización** para evitar posibles inconsistencias en la base de datos

Normalización

- Un buen diseño evita determinados problemas, ya que el objetivo es que el sistema se encargue, de forma automática, de evitar inconsistencias.

- La normalización ayuda a obtener un buen diseño:
  - Expresar restricciones sobre los datos
  - Uso de estas restricciones para descomponer relaciones

- Las relaciones obtenidas estarán en alguna "forma normal" que garantizará que se cumplen ciertas propiedades.

Normalización

EmpleadosCentro

Id	Nombre	Direccion	Categ	Salario	Centro	DirCentro	Tfno
1Y	Ana	Alcala 20	Profesor	25	001	Complutense	9658
4B	Benito	Arroyo 122	Auxiliar	15	002	Somosaguas	7855
7L	Carlos	Lopera 478	Ayudante	18	001	Complutense	9658
6P	David	Salud 41	Técnico	21	003	Complutense	2545

- Redundancia de los datos asociados a cada centro.
- Anomalías de inserción:
  - Se podría añadir un empleado adscrito al centro 001 con un teléfono diferente a 9658.
  - Solo se puede insertar un nuevo centro si existe un empleado destinado en él.

Normalización

EmpleadosCentro

Id	Nombre	Direccion	Categ	Salario	Centro	DirCentro	Tfno
1Y	Ana	Alcala 20	Profesor	25	001	Complutense	9658
4B	Benito	Arroyo 122	Auxiliar	15	002	Somosaguas	7855
7L	Carlos	Lopera 478	Ayudante	18	001	Complutense	9658
6P	David	Salud 41	Técnico	21	003	Complutense	2545

- Anomalías de modificación:
  - Modificación del teléfono del centro 001 en la fila del empleado 1Y.
- Anomalías de eliminación
  - Si se da de baja el empleado 4B se perderán los datos del centro 002.

Descomposición

- Muchos problemas que surgen con la redundancia se pueden resolver sustituyendo la relación dada por un conjunto de relaciones menores: *Descomposición del esquema relación.*

- *Las Dependencias Funcionales y las Formas Normales*

  proporcionan criterios y mecanismos para decidir si una relación debe ser descompuesta o no, lo que lleva a una redefinición del esquema de la base de datos

Normalización

- Las formas normales son una serie de reglas cuyo cumplimiento asegura que el esquema diseñado tenga un buen comportamiento en cuanto a redundancia, pérdida de información...
- Estas reglas se basan en las dependencias funcionales.
- Una dependencia funcional es una restricción de integridad que generaliza la noción de superclave.

Dependencias Funcionales

- Es una propiedad semántica de un esquema de relación que determina para cada valor de un conjunto de atributos X el valor de otro conjunto de atributos Y.

  - Dados $X, Y \subseteq R$ una dependencia funcional $X \longrightarrow Y$ especifica

    $\forall\, t1, t2 \in R$ tal que $t1[X] = t2[X] \Longrightarrow t1[Y] = t2[Y]$

  - Los valores de X determinan unívocamente los valores de Y.

  - Todas las tuplas que tienen los mismos valores en los atributos del conjunto X también deben tener los mismos valores en los atributos del conjunto Y.

Dependencias Funcionales

Id	Nombre	Direccion	Categ	Salario	Centro	DirCentro	Tfno
1Y	Ana	Alcala 20	Profesor	25	001	Complutense	9658
4B	Benito	Arroyo 122	Auxiliar	15	002	Somosaguas	7855
7L	Carlos	Lopera 478	Ayudante	18	001	Complutense	9658
6P	David	Salud 41	Técnico	21	003	Complutense	2545

- $\{id\} \longrightarrow \{Nombre, Dirección, Categ, Salario, Centro, DirCentro, Tfno\}$
- $\{Centro\} \longrightarrow \{DirCentro, Tfno\}$
- $\{Tfno\} \longrightarrow \{Centro\}$

- Sin embargo no se cumple $\{DirCentro\} \longrightarrow \{Centro\}$

Dependencias Funcionales

- $X \longrightarrow Y$ no implica necesariamente $Y \longrightarrow X$

  $\{Nif\} \longrightarrow \{Nombre\}$ pero no es cierto que $\{Nombre\} \longrightarrow \{Nif\}$

- Una superclave se puede definir en términos de dependencias funcionales.

  S es superclave de un esquema de relación R si $S \subseteq R$ y $S \longrightarrow R$

- Algunas dependencias funcionales son triviales porque las satisfacen todas las relaciones.

  $S \longrightarrow S$ la satisfacen todas las relaciones que contengan al conjunto de atributos S

- En general, una dependencia funcional de la forma $S \longrightarrow R$ es trivial si $R \subseteq S$.

Dependencias Funcionales

- Dado un cierto conjunto D de dependencias funcionales, es deseable encontrar otro conjunto E que sea lo menor posible de manera que cada $d \in D$ se deduzca de E, con el objetivo de que el coste de mantener la integridad definida en D se reduzca con E.

- Una de las maneras de reducir el coste es *eliminar las dependencias que no aportan nada semánticamente.*

Dependencias Funcionales

- El cierre de un conjunto de dependencias funcionales S , denotado S+ , es el conjunto de todas las dependencias funcionales que se pueden deducir intensionalmente de S.
- Para calcular el cierre de un conjunto de dependencias funcionales se dispone de un conjunto de axiomas de producción denominados Axiomas de Armstrong.

1. Reflexividad: Si $Y \subseteq X$, entonces $X \longrightarrow Y$
2. Aumentatividad: Si $X \longrightarrow Y$ entonces $XZ \longrightarrow YZ$
3. Transitividad: Si $X \longrightarrow Y$ y $Y \longrightarrow Z$ entonces $X \longrightarrow Z$

Dependencias Funcionales

- Hay otras reglas de inferencia que se deducen de los axiomas de Armstrong y que permiten calcular mas rápidamente el cierre de un conjunto de dependencias funcionales:
  1. Unión: Si $X \longrightarrow Y$ y $X \longrightarrow Z$ entonces $X \longrightarrow YZ$
  2. Descomposición: Si $X \longrightarrow YZ$ entonces $X \longrightarrow Y$ y $X \longrightarrow Z$
  3. Composición: Si $X \longrightarrow Y$ y $Z \longrightarrow W$ entonces $XZ \longrightarrow YW$
  4. Pseudotransitividad: Si $X \longrightarrow Y$ y $YZ \longrightarrow W$ entonces $XZ \longrightarrow W$

Dependencias Funcionales

- Dado el conjunto S de dependencias funcionales:
  - $A \rightarrow BC$
  - $CD \rightarrow EF$
- Se puede demostrar que $AD \rightarrow F$, está en S^+ :
  - $A \rightarrow BC$
  - $A \rightarrow C$ (descomposición)
  - $AD \rightarrow CD$ (aumentatividad)
  - $CD \rightarrow EF$ (dada)
  - $AD \rightarrow EF$ (transitividad)
  - $AD \rightarrow F$ (descomposición)

Dependencias Funcionales

- En la práctica no es necesario en general calcular todo el cierre de un conjunto de dependencias.
- Es mas interesante *calcular el conjunto de las dependencias que tienen en su parte izquierda un conjunto especifico de atributos.*
  1. Comprobar si una DF se deduce de un conjunto de DFs. Se puede determinar si su comprobación es redundante para la integridad de los datos.
  2. Comprobar si un conjunto de atributos es superclave. Asegura que un conjunto de atributos es adecuado para determinar unívocamente cada tupla de una relación.
  3. Calcular un conjunto mínimo de DFs. Mantiene la comprobación de integridad menos costosa.

Dependencias Funcionales

- El cierre de un conjunto de atributos X con respecto a un conjunto de dependencias funcionales S, denotado X_S^+ , es el conjunto de atributos Y tales que $X \rightarrow Y$ se puede deducir de S.
- Algoritmo

  Entrada: Conjunto de atributos X y un conjunto de DFs S.

  Salida: X_S^+

  resultado := X

  while cambios en resultado do

  for each $Y \rightarrow Z \in S$ do

  if $Y \subseteq$ resultado then resultado := resultado $\cup$ Z.

Dependencias Funcionales

- Sea el conjunto de dependencias funcionales

  $$\{A \rightarrow B, A \rightarrow C,\ CG \rightarrow H, CG \rightarrow I, B \rightarrow H\}$$

- Calculo de $(AG)+$
  - Resultado = AG
  - $A \rightarrow B$. Resultado = AGB
  - $A \rightarrow C$. Resultado = $AGBC$
  - $CG \rightarrow H$. Resultado = $AGBCH$
  - $CG \rightarrow I$. Resultado = $AGBCHI$
  - $B \rightarrow H$. Resultado = $AGBCHI$

  - En el siguiente bucle no varía el resultado

Dependencias Funcionales

- El cierre de un conjunto de atributos X con respecto a un conjunto de dependencias funcionales S, denotado X_S^+ , es el conjunto de atributos Y tales que $X \rightarrow Y$ se puede deducir de S.

  - $X \rightarrow Y$ se deduce de un conjunto de dependencias funcionales $S \Leftrightarrow Y \subseteq X_S^+$
  - Un conjunto de atributos C es superclave de una relación R bajo un conjunto de dependencias funcionales S si todos los atributos de R pertenecen a C_S^+.
  - Además, será clave candidata si el conjunto de atributos C es irreducible.

Dependencias Funcionales

- Dados dos conjuntos de dependencias funcionales S_1 y S_2, se dice que S_2 es un recubrimiento de S_1 si cada dependencia de S_1 se deduce de S_2 (es decir, se puede demostrar que cada dependencia de S_1 esta en el cierre de S_2).

- Dos conjuntos de dependencias funcionales S_1 y S_2 son equivalentes si $S_1+ = S_2+$

- Al conjunto mínimo de dependencias funcionales (no presenta dependencias funcionales redundantes) equivalente a S se le denomina recubrimiento mínimo de S.

Dependencias Funcionales

- Un conjunto S de dependencias funcionales es irreducible si y solamente si cumple las siguientes propiedades:
  - La parte derecha de cada dependencia funcional de S tiene solo un atributo.
  - La parte izquierda de cada dependencia funcional de S es irreducible en el sentido en que si se elimina algún atributo, necesariamente cambia el cierre de S.
  - No se puede eliminar ninguna dependencia funcional de S sin cambiar su cierre.
- Se puede demostrar que todo conjunto de dependencias funcionales tiene al menos un recubrimiento mínimo

Dependencias Funcionales

- Entrada: Conjunto S de dependencias funcionales
- Salida: Sm recubrimiento mínimo de S

1. Dejar lados derechos unitarios. Sea $Sm = \emptyset$. Por cada dependencia $X \longrightarrow A_1 ... A_n$ con A_i atributos, incluir en Sm las n dependencias $X \longrightarrow A_1 ... X \longrightarrow A_n$
2. Eliminar atributos redundantes. Para cada dependencia funcional $X \longrightarrow Y \in Sm$, $X = \{A_1 ... A_n\}$ y para cada $i = 1 ... n$, comprobar si $(X - A_i) \longrightarrow Y \in Sm+$. A_i es redundante en la dependencia $(Y \subseteq (X - A_i)^+)$ y debemos eliminarlo

   $$Sm = Sm - \{X \longrightarrow Y\} \cup \{X - A_i \longrightarrow Y\}$$

3. Eliminar dependencias redundantes. Para cada dependencia funcional $X \longrightarrow Y \in Sm$ comprobar si $X \longrightarrow Y \in (Sm - \{X \longrightarrow Y\}) + (Y \subseteq X^+$ en $Sm - \{X \longrightarrow Y\})$. En ese caso $X \longrightarrow Y$ es redundante y debemos eliminarla: $Sm = Sm - \{X \longrightarrow Y\}$

Dependencias Funcionales

- Ejercicio: Encontrar un recubrimiento mínimo para el siguiente conjunto de dependencias funcionales S

$$S = \{A \rightarrow BC, B \rightarrow C, A \rightarrow B, AB \rightarrow C, AC \rightarrow D\}$$

- $Sm = \{A \longrightarrow B, A \longrightarrow C, B \longrightarrow C, AB \longrightarrow C, AC \longrightarrow D\}$
- $AB \longrightarrow C \qquad C \in \{A\}^+ = \{A, B, C, D\}$

  $Sm = \{A \longrightarrow B, A \longrightarrow C, B \longrightarrow C, AC \longrightarrow D\}$
- $AC \longrightarrow D \qquad D \notin \{C\}^+ = \{C\} \quad D \in \{A\}^+ = \{A, B, C, D\}$

  $Sm = \{A \longrightarrow B, A \longrightarrow C, B \longrightarrow C, A \longrightarrow D\}$

Dependencias Funcionales

- Ejercicio: Encontrar un recubrimiento mínimo para el siguiente conjunto de dependencias funcionales S

$$Sm = \{A \longrightarrow B, A \longrightarrow C, B \longrightarrow C, A \longrightarrow D\}$$

- $A \longrightarrow B \quad B \notin \{A\}^+ = \{A, C, D\}$
- $A \longrightarrow C \quad C \in \{A\}^+ = \{A, B, C, D\}$ Se puede eliminar

  $S = \{A \longrightarrow B, B \longrightarrow C, A \longrightarrow D\}$
- $B \longrightarrow C \quad C \notin \{B\}^+ = \{B\}$
- $A \longrightarrow D \quad D \notin \{A\}^+ = \{A, B, C\}$

$$Sm = \{A \longrightarrow B, B \longrightarrow C, A \longrightarrow D\}$$

Forma Normal

- La **forma normal** de una relación corresponde a la mayor condición de forma normal que satisface un esquema de relación, indicando así el grado hasta el que se ha normalizado.
- Lograremos la normalización mediante descomposiciones.
- Las formas normales básicas son
  - Primera (1FN)
  - Segunda (2FN)
  - Tercera (3FN)
  - Boyce/Codd (FNBC)

Forma Normal

- En general, los diseños prácticos exigen 3FN, aunque hay circunstancias especiales en las que 2FN puede ser suficiente.
- Hay otras formas normales mas exigentes.
- El proceso de asegurar una forma normal para un esquema se denomina **normalización.**

Primera Forma Normal

- Se considera parte de la definición formal de relación, porque establece:

Los dominios de los atributos solo pueden ser atómicos, para evitar atributos multivalorados, compuestos y sus combinaciones.

Primera Forma Normal

- Si se asume que un centro puede tener mas de un teléfono, podríamos tener la siguiente representación

Centro	DirCentro	Tfno
001	Complutense	{9658,2545,1458}
002	Somosaguas	{7855}
003	Complutense	{3215,4785}

- Hay tres posibilidades de representar la entidad para satisfacer la primera forma normal.

Primera Forma Normal

Eliminar el atributo Tfnos y crear una nueva relación que asocie a cada centro diferentes telefonos.

Centro	DirCentro
001	Complutense
002	Somosaguas
003	Complutense

Centro	Tfno
001	9658
001	1458
002	7855
001	2545
003	3215
003	4785

Primera Forma Normal

Ampliar la clave de la relación de manera que incluya al atributo multivalorado. Presenta como inconveniente redundancia que provoca anomalías

Centro	Direccion	Tfno
001	Complutense	9658
001	Complutense	1458
002	Somosaguas	7855
001	Complutense	2545
003	Complutense	3215
003	Complutense	4785

Primera Forma Normal

Si se conoce la cardinalidad máxima del atributo multivalorado, se pueden crear tantos atributos como la cardinalidad máxima. Presenta como inconveniente el uso de valores NULL.

Centro	DirCentro	Tfno1	Tfno2	Tfno3
001	Complutense	9658	1458	2545
002	Somosaguas	7855		
003	Complutense	3215	4785	

Segunda Forma Normal

- Se basa en el concepto de dependencia funcional total.
- Un atributo se dice primo si es miembro de alguna clave candidata.
- Una dependencia funcional $X \longrightarrow Y$ es una dependencia funcional parcial si existe $Z \subset X$ tal que $Z \longrightarrow Y$. Cuando no se cumple se dice que se trata de una dependencia funcional total.

Todos los atributos no primos dependen funcionalmente de forma total de las claves candidatas.

Segunda Forma Normal

- Existen anomalías de actualización causadas por las dependencias df_2 y df_3.

$$df_1 = \{IdEmp, IdProy\} \longrightarrow \{Horas\}$$
$$df_2 = \{IdEmp\} \longrightarrow \{NombreE\}$$
$$df_3 = \{IdProy\} \longrightarrow \{NombreP\}$$

IdEmp	IdProy	Horas	NombreE	NombreP
1256	P1	8	Pablo Almeida	Gestión Nominas
0145	P2	12	Susana García	Planificación
0145	P1	6	Susana García	Gestión Nominas

Segunda Forma Normal

- La segunda forma normal evita este tipo de anomalías.
- Un esquema que no se encuentre en segunda forma normal puede traducirse en varios esquemas que si lo estén, creando tantas nuevas relaciones como dependencias funcionales no sean completas

$$\{Ai \mid Ai \in Clave\ Candidata\} \longrightarrow \{Atributo\ no\ primo\}$$

PE1 (IdEmp, IdProy, Horas)

PE2 (IdEmp, NombreE)

PE3 (IdProy, NombreP)

Tercera Forma Normal

- La tercera forma normal se basa en el concepto de dependencia funcional transitiva

- Una dependencia funcional $X \longrightarrow Y$ es una dependencia funcional transitiva si existe un conjunto de atributos Z que ni es clave candidata ni es subconjunto de ninguna clave y además se cumple $X \longrightarrow Z$ y $Z \longrightarrow Y$.

Ninguno de los atributos no primos depende transitivamente de una clave candidata.

Tercera Forma Normal

- Un esquema está en tercera forma normal con respecto a un conjunto de dependencias funcionales S si para toda dependencia funcional $X \rightarrow Y$ de S^+ no trivial se cumple que
  - o bien X es superclave
  - o bien Y forma parte de una clave candidata
- El procedimiento para normalizar esta relación consiste en descomponerla en los atributos definidos por la dependencia funcional responsable de la transitividad.

Tercera Forma Normal

IdEmp	NombreE	CodDep	NombreDep	DireccionDep
1256	Pablo Almeida	D234	RRHH	0325
0145	Susana García	D234	RRHH	0325
0145	Alberto Gómez	D567	Personal	0897

$$df_1 = \{IdEmp\} \longrightarrow \{NombreE, CodDep\}$$
$$df_2 = \{CodDep\} \longrightarrow \{NombreDep, DireccionDep\}$$

Dependencia Funcional Transitiva

$$df_3 = \{IdEmp\} \longrightarrow \{NombreDep, DireccionDep\}$$

ED1 (CodDep, NombreDep, DireccionDep)

ED2 (IdEmp, NombreE, CodDep)

Forma normal de Boyce-Codd

Un esquema está en forma normal de Boyce-Codd, con respecto a un conjunto de dependencias funcionales S, si para toda dependencia funcional no trivial $X \rightarrow Y$ de S^+ se cumple que X es superclave

- La forma normal de Boyce-Codd (FNBC) es mas estricta que la 3FN, aunque su definición es mas simple.

Forma normal de Boyce-Codd

Investigador	Proyecto	IdPrincipal
111G	Proyecto1	01458P
222F	Proyecto1	01259T
333Y	Proyecto1	01458P
444P	Proyecto2	78978M
444P	Proyecto1	01458P

$$df_1 = \{Investigador, Proyecto\} \longrightarrow \{IdPrincipal\}$$
$$df_2 = \{IdPrincipal\} \longrightarrow \{Proyecto\}$$

- Si no se vigila la dependencia funcional df_2 se podría añadir una tupla de modo que un investigador fuese investigador principal de más de un proyecto.
- Hay diferentes descomposiciones posibles.

Forma normal de Boyce-Codd

- Opción 1
  - P1A(Idprincipal, Proyecto)
  - P2A(Investigador, Proyecto)
- Opción 2
  - P1B(Idprincipal, Proyecto)
  - P2B(Investigador, Idprincipal)

- Estas descomposiciones pierden la dependencia funcional df_1 porque estas se refieren al contexto local de un esquema, no hacen referencia a esquemas externos.

Forma normal de Boyce-Codd

IdPrincipal	Proyecto
01458P	Proyecto1
01259T	Proyecto1
01458P	Proyecto1
78978M	Proyecto2
01458P	Proyecto1

Investigador	Proyecto
111G	Proyecto1
222F	Proyecto1
333Y	Proyecto1
444P	Proyecto2
444P	Proyecto1

- Se generan tuplas incorrectas en la operación join.

IdPrincipal	Proyecto	Investigador
01458P	Proyecto1	111G
01458P	Proyecto1	222F
01458P	Proyecto1	333Y
01458P	Proyecto1	444P
…	…	…

Forma normal de Boyce-Codd

- Es necesario encontrar el modo de crear descomposiciones que, como mínimo, no generen tuplas incorrectas en la reunión.

- Si las dependencias funcionales no se aseguran en la descomposición, como en el ejemplo anterior, el diseñador debe tenerlas presente y asegurarlas por programación.

Preservación de dependencias funcionales

- Para poder determinar formalmente la condición de preservación de dependencias en la descomposición introducimos el concepto de proyección de un conjunto de dependencias funcionales.

- La proyección de un conjunto de dependencias funcionales S sobre los atributos R_i, denotada por $\Pi_{R_i}(S)$, con $R_i \subseteq R$ y R esquema de relación, es el conjunto de dependencias funcionales $X \longrightarrow Y$ en S tales que $X \cup Y \subseteq R_i$.

Preservación de dependencias funcionales

- Una descomposición $D = \{R_1 ... R_n\}$ del esquema R preserva las dependencias funcionales con respecto al conjunto de dependencias funcionales S sobre R si la unión de las proyecciones de S sobre cada R_i es equivalente a S, es decir:

$$\left(\Pi_{R_1}(S) \cup \cdots \cup \Pi_{R_n}(S)\right)+ = S +$$

- Siempre es posible encontrar una descomposición D que preserve las dependencias con respecto a S tal que cada relación R_i este en 3FN.

Preservación de dependencias funcionales

- Entrada: Un esquema R y un conjunto de dependencias funcionales S.
- Salida: Una descomposición D que preserve las dependencias con respecto a S en 3FN.
  1. Encontrar un recubrimiento mínimo T para S.
  2. Para cada $X \longrightarrow Y \in T$ crear un esquema en D con atributos $X \cup \{A_1\} \cup \cdots \cup \{A_n\}$, donde $X \longrightarrow \{A_1\}, \ldots, X \longrightarrow \{A_n\}$ son todas las dependencias funcionales en T cuya parte izquierda es X. (Por tanto, X es clave para el nuevo esquema).
  3. Poner el resto de atributos en una única relación para asegurar la preservación de atributos.

Descomposición y reuniones no aditivas

- Una propiedad necesaria de mantener en la descomposición es que no se creen tuplas con valores incorrectos cuando una descomposición se reúne con la operación join. La no aditividad se refiere a no producir más tuplas de las que estaban en la relación antes de la descomposición.

- Una descomposición $D = \{R_1 ... R_n\}$ del esquema R presenta la propiedad de reunión no aditiva con respecto al conjunto de dependencias funcionales S sobre R si para todo estado r de la relación R que satisfaga S, se cumple:

$$* \left(\Pi_{R_1}(r), \ldots, \Pi_{R_n}(r)\right) = r$$

donde * representa el natural Join de las relaciones.

Descomposición y reuniones no aditivas

- Una descomposición binaria $D = \{R_1, R_2\}$, del esquema R presenta la propiedad de reunión no aditiva con respecto al conjunto de dependencias funcionales S sobre R si y solo si se cumple alguna de las dos condiciones siguientes:

$$\big((R_1 \cap R_2) \longrightarrow (R_1 - R_2)\big) \in S+$$
$$\big((R_1 \cap R_2) \longrightarrow (R_2 - R_1)\big) \in S+$$

Descomposición y reuniones no aditivas

- Descomposiciones sucesivas
  - Una descomposición $D = \{R_1 ... R_n\}$ del esquema R presenta la propiedad de reunión no aditiva con respecto al conjunto de dependencias funcionales S sobre R , y
  - Una descomposición $D' = \{Q_1 ... Q_s\}$ del esquema Ri presenta la propiedad de reunión no aditiva con respecto a $\Pi_{R_i}(S)$, entonces la descomposición

$$D = \{R_1 ... R_{i-1,}\, Q_1 ... Q_s, R_{i+1} ... R_n\}$$

también presenta la propiedad de reunión no aditiva con respecto al conjunto de dependencias funcionales S sobre R.

Descomposición y reuniones no aditivas

- Entrada: Un esquema R y un conjunto de dependencias funcionales S sobre R.
- Salida: Una descomposición en FNBC que preserva la propiedad de reunión no aditiva.

1. $D := \{R\}$
2. while existe $Q \in S$, tal que Q no esté en FNBC do

   encontrar una dependencia funcional $X \longrightarrow Y$ *de* S *que viole la FNBC;*

   reemplazar R *por dos esquemas de relación* $(Q - Y)$ *y* $(X \ U \ Y)$

Descomposición y reuniones no aditivas

- Si se desean preservar las dependencias en la descomposición y reuniones no aditivas, en general solo se puede asegurar que el resultado se encuentre en 3FN
- Entrada: Un esquema R y un conjunto de dependencias funcionales S sobre R.
- Salida: Una descomposición en 3FN que preserva dependencias y reuniones no aditivas.
  1. Encontrar un recubrimiento mínimo T para S.
  2. Para cada $X \longrightarrow Y \in T$ crear un esquema en D con atributos $\{X, A_1, \ldots, A_n\}$ donde $X \longrightarrow A_1, \ldots, X \longrightarrow A_n$ son todas las dependencias funcionales en T cuya parte izquierda es X.
  3. Si ninguno de los esquemas contiene una clave candidata, crear un nuevo esquema que la contenga.