



**Examen de  
Bases de datos y sistemas de información  
I PARCIAL**

- 1) (0,5 puntos) Demostrar que en la tabla que resulta de traducir un conjunto de relaciones  $n$ -arias en el que sólo  $m$  conjuntos de entidades participan con cardinalidad (1,1) y el resto con cardinalidad  $N$  al modelo relacional puede haber a lo sumo  $m$  claves candidatas. ¿Cuándo hay exactamente  $m$ ? ¿Qué ocurre si la restricción de cardinalidad es (0,1)?

R: Para cada conjunto de entidades que participan con cardinalidad máxima 1, sus valores de clave sólo pueden aparecer una vez en la tabla del conjunto de relaciones, por lo que su clave puede formar superclave. Serán claves candidatas si no hay superclaves con menos atributos.

Si todos estos conjuntos de entidades tienen el mismo número de atributos clave, habrá  $m$  claves candidatas (no puede haber más porque los atributos clave de los conjuntos de entidades que participen con cardinalidad superior no pueden formar clave de la tabla del conjunto de relaciones, ya que sus valores aparecerán en general más de una vez).

Cuando la restricción de cardinalidad es (0,1), los atributos clave de estos conjuntos de entidades pueden seguir siendo clave candidata de la tabla del conjunto de relaciones. La diferencia con el caso anterior se refiere a la cardinalidad de los estos conjuntos de entidades. Mientras que en el caso anterior (cardinalidad (1,1), es decir, participación total) todas las entidades están representadas en el conjunto de relaciones, en este caso no. Por lo que en el primero, la cardinalidad de estos conjuntos de entidades debe coincidir, mientras que en el segundo no.

- 2) (1 punto) Expresar la operación de reunión zeta externa completa de dos relaciones  $r$  y  $s$ , con esquemas respectivos  $R$  y  $S$ , en términos de operaciones del álgebra relacional asumiendo que no se tiene ningún tipo de reunión externa.

R:  $(r \times |x|_q s) \cup ((\{nulo, \dots, nulo\} \times (s - \Pi_S(r|x|_q s))) \cup ((r - \Pi_R(r|x|_q s)) \times \{nulo, \dots, nulo\}))$

donde la primera relación  $\{nulo, \dots, nulo\}$  se encuentra en el esquema  $R-S$  y la segunda en el esquema  $S-R$ .

- 3) (0,5 puntos) Supóngase que se realiza una reunión entre dos tablas A y B y que el resultado contiene columnas de ambas. Observando el resultado, ¿qué tipos de relaciones según su cardinalidad puede haber entre las columnas de A y B en el resultado?

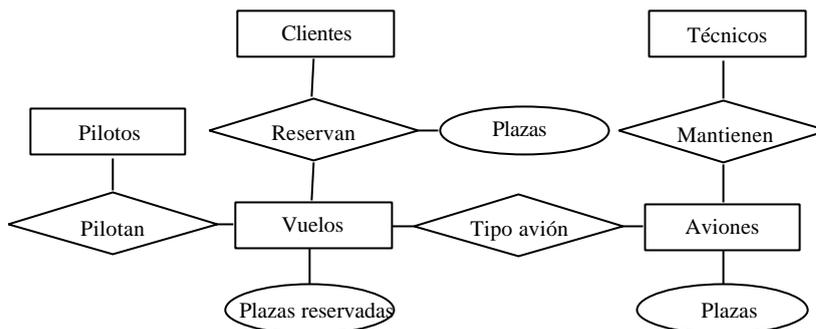
R:

- Las reuniones que encajan las claves primarias con las externas siempre producen relaciones uno a varios.
  - Otras reuniones también pueden generar relaciones uno a varios si las columnas coincidentes en al menos una de las tablas tiene valores únicos en todas las filas de la tabla.
  - Si las columnas coincidentes tienen valores únicos en todas las filas ambas tablas se trata de relaciones uno a uno.
  - En general, las reuniones de columnas coincidentes arbitrarias generan relaciones varios a varios.
- 4) (1,5 puntos) Dado el siguiente modelo entidad relación (se asume que todas las entidades están determinadas por códigos y un coste de escritura doble al de lectura) y la siguiente tabla de volúmenes de datos, se pide determinar la eliminación o no de la información redundante con respecto a las operaciones:
- a) Operación 1: Hacer una reserva en un vuelo (frecuencia: 50 veces al día).
  - b) Operación 2: Listar todos los vuelos con el tipo de avión, su número de plazas, reservas y nombres de los técnicos que los mantienen (frecuencia: 10 veces al día)



Tabla de volúmenes de datos

Concepto	Tipo	Volumen
Clientes	E	10.000
Vuelos	E	1.000
Aviones	E	200
Pilotos	E	300
Técnicos	E	1.600
Reservan	R	20.000
Pilotan	R	1.000
Tipo avión	R	1.000



R:

Sólo hay una redundancia: el número de plazas reservadas (Plazas reservadas) en Vuelos, que se puede determinar mediante la relación Reservan.

Requisito de almacenamiento para el atributo redundante:  $4 \times 1.000 = 4.000$  bytes (asumiendo un tipo de datos entero con cuatro bytes de almacenamiento).

Cada avión está mantenido en media por 8 técnicos ( $1.600/200$ ).

Se asume que los clientes ya están dados de alta en la base de datos.

**Primer caso: con redundancia.**

Coste de las operaciones:

Tabla de accesos

Operación 1				
Concepto	Tipo de objeto	Accesos	Tipo de acceso	Motivo
Vuelos	E	1	L	Para leer el número de plazas reservadas
Tipo avión	R	1	L	Para obtener el tipo de avión que le corresponde
Aviones	E	1	L	Para determinar el número de plazas del avión
Reservan	R	1	E	Para insertar la nueva tupla de Reservan (si quedan plazas)
Vuelos	E	1	E	Para escribir el nuevo número de plazas reservadas (si quedan plazas)
Operación 2				
Concepto	Tipo	Accesos	Tipo	Motivo
Vuelos	E	1x1000	L	Para obtener el número de plazas reservadas en el vuelo y su código
Tipo avión	R	1x1000	L	Para obtener el tipo de avión que le corresponde
Aviones	E	1x1000	L	Para determinar el número de plazas del avión
Mantienen	R	8x1000	L	Para obtener los técnicos que mantienen el avión
Técnicos	E	8x1000	L	Para determinar el nombre de cada uno de los técnicos

Leyenda: E=Entidad, R=Relación, L=Lectura, E=Escritura

Operación 1:  $3 \times 50 = 150$  accesos en lectura y  $2 \times 50 = 100$  en escritura. Coste =  $150 + 100 \times 2 = 350$

Operación 2:  $19.000 \times 10 = 190.000$  accesos en lectura. Coste =  $190.000$

Coste total:  $350 + 190.000 = 190.350$

**Segundo caso: sin redundancia.**

En media, en cada vuelo reservan 20 clientes ( $20.000/1.000$ )

Tabla de accesos

Operación 1				
Concepto	Tipo de objeto	Accesos	Tipo de acceso	Motivo
Tipo avión	R	1	L	Para obtener el tipo de avión que corresponde al vuelo
Aviones	E	1	L	Para determinar el número de plazas del avión
Reservan	R	20 (en media)	L	Para determinar el número de plazas reservadas
Reservan	R	1	E	Para insertar la nueva tupla de Reservan (si quedan plazas)



Operación 2				
Concepto	Tipo	Accesos	Tipo	Motivo
Vuelos	E	1x1000	L	Para obtener el número de plazas reservadas en el vuelo y su código
Tipo avión	R	1x1000	L	Para obtener el tipo de avión que le corresponde
Aviones	E	1x1000	L	Para determinar el número de plazas del avión
Mantienen	R	8x1000	L	Para obtener los técnicos que mantienen el avión
Técnicos	E	8x1000	L	Para determinar el nombre de cada uno de los técnicos
Reservan	R	20x1000	L	Para determinar el número de reservas

Leyenda: E=Entidad, R=Relación, L=Lectura, E=Escritura

Operación 1:  $22 \times 50$  accesos en lectura y  $1 \times 50$  en escritura. Coste =  $1.100 + 50 \times 2 = 1.200$

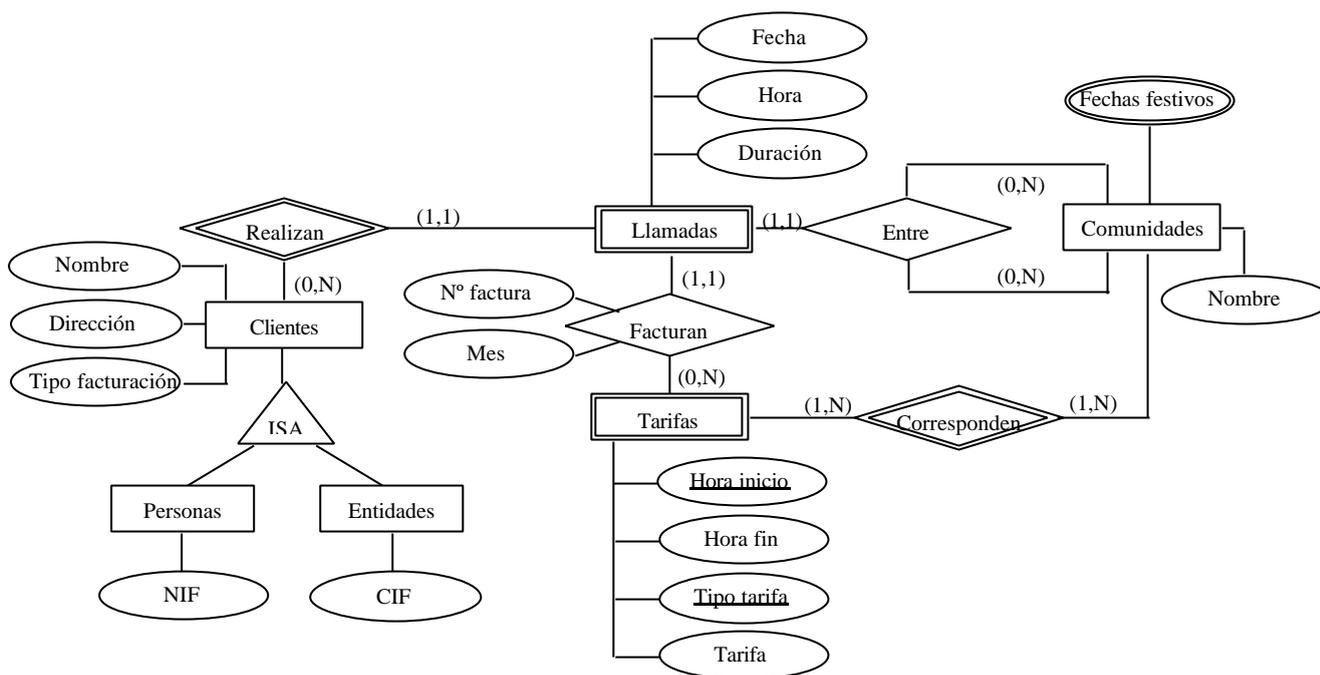
Operación 2:  $39.000 \times 10 = 390.000$  accesos en lectura. Coste =  $390.000$

Coste total:  $1.200 + 390.000 = 391.200$

Por tanto, se escoge con redundancia.

- 5) (3,5 puntos) Un operador de telefonía celular desea contar con una base de datos que guarde información sobre las llamadas de sus clientes y su facturación. Los clientes pueden ser personas individuales (con un NIF) o entidades (con un CIF). Cada cliente elige el tipo de facturación que desea (mensual o bimensual) y la dirección de envío de las facturas, cada una de ellas identificada por un número de factura. La facturación de cada llamada depende de su duración, sus comunidades autónomas de origen y destino, y de unas tarifas fijas para todos los clientes que distinguen las realizadas en festivos del resto (no hay cuota de establecimiento de llamada). Estas tarifas están definidas por intervalos de tiempo para las llamadas en días festivos y laborables y no se contempla el IVA. Se pide:
- Construir el modelo entidad-relación añadiendo el mínimo número de atributos, con generalizaciones donde sea posible e imponiendo y explicando las restricciones de clave, de dominio y de cardinalidad mínimo-máximo que se estimen oportunas.
  - Traducir el modelo conceptual obtenido al modelo lógico sin considerar las operaciones expresando las restricciones de integridad referencial y de participación total en notación algebraica.
  - Plantear una consulta en SQL para obtener el coste de una llamada, asumiendo que las llamadas entre diferentes comunidades tarifan el doble de las que se realicen dentro de la misma.
  - Plantear una consulta en SQL para emitir la factura de un cliente para un mes determinado con los datos de cada llamada (fecha, hora, origen, destino, coste) y otra que calcule el total de la factura.

R. a)



a1) Restricciones de clave y de dominio

*Generalizaciones:*

Clientes(Nombre: String, Dirección: String, Tipo facturación: { Mensual, Bimensual })

*Entidades fuertes:*

Personas(NIF: String)

Entidades(CIF: String)

Comunidades(Nombre: String)

*Entidades débiles:*

Llamadas(Fecha: Date, Hora: Time, Duración: Integer, Clientes.Identificación)

Identificación es el valor de NIF de Personas o de CIF de Entidades, según corresponda

Festivos(Fecha: Date, Comunidades.Nombre)

Tarifas(Hora inicio: Time, Hora fin: Time, Tipo tarifa: { Laborable, Festivo }, Tarifa: Numeric(5,4), Comunidades.Nombre)

a2) Restricciones de cardinalidad mínimo-máximo

Realizan:

- Clientes - (0,N) - Realizan - Llamadas: No todos los clientes dados de alta deben haber efectuado una primera llamada (mínimo 0). El mismo cliente puede haber realizado varias llamadas (máximo N).
- Clientes - Realizan - (1,1) Llamadas: Cada llamada debe pertenecer a un cliente (mínimo 1). Una llamada corresponde sólo a un cliente (máximo 1)

Facturan:

- Llamadas - (1,1) - Facturan - Tarifas: Toda llamada debe pertenecer a una factura (mínimo 0). Cada llamada pertenece a una única factura (máximo 1).
- Llamadas - Facturan - (0,N) - Tarifas: Puede haber tarifas para las que no exista ninguna llamada (mínimo 0). Para una tarifa puede haber varias llamadas (máximo N).

Corresponden:

- Tarifas - (1,N) - Corresponden - Comunidades: Toda tarifa debe corresponder a una comunidad (mínimo 1). La misma tarifa puede corresponder a varias comunidades (máximo N).
- Tarifas - Corresponden - (1,N) - Comunidades: Cada comunidad tiene al menos una tarifa (mínimo 1). Cada comunidad puede tener más de una tarifa (máximo N).

Tienen:

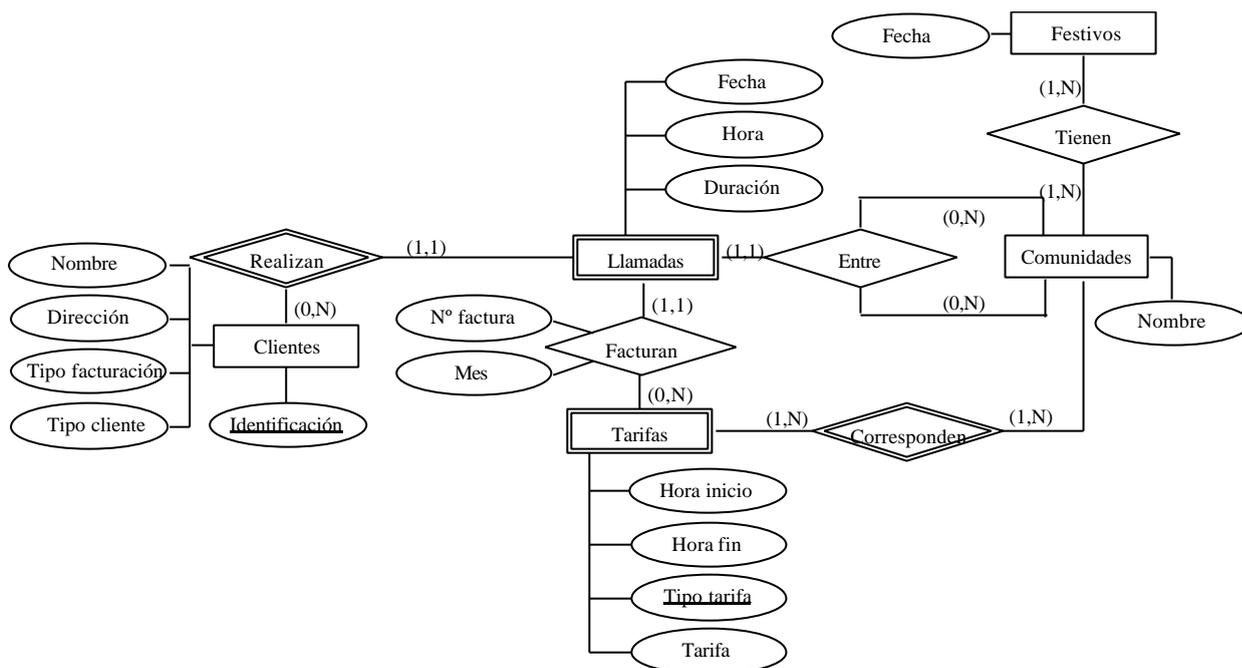
- Comunidades - (1,N) - Tienen - Festivos: Cada comunidad tiene al menos un festivo (mínimo 1). Cada comunidad puede tener varios festivos (máximo N).
- Comunidades - Tienen - (1,N) - Festivos: Cada festivo debe estar asociado a una comunidad (mínimo 1). Un festivo en concreto puede corresponder a varias comunidades (máximo N); por ejemplo, los días festivos nacionales.

b)

**Fase de reestructuración:**

Sólo se eliminarán las generalizaciones, no se consideran divisiones ni mezclas de entidades y relaciones porque no se tienen en cuenta las operaciones. Se transformará el único atributo multivalorado (Fechas festivos) en un conjunto de entidades (Festivos) relacionado con Comunidades a través de la relación Tienen.

La única generalización se transforma con un plegado de las entidades hijo con la entidad padre. Se conserva una sola relación entre Clientes y Llamadas (a diferencia de las otras dos alternativas posibles, que generarían dos relaciones). Aparece un nuevo atributo en Clientes, Tipo cliente, que denotará el tipo de cliente, con dominio {Persona, Entidad}. Dado que los dominios de NIF y CIF son iguales, se puede usar un único atributo para denotar a ambos, Identificación. Con ello no aparecerán nulos. Sin embargo, hay que ser consciente de que NIF y CIF deberían tener conceptualmente dominios diferentes allá donde quepa hacer esta distinción explícita. No obstante, no parece que sea necesario en esta aplicación. Así:

**Traducción al modelo lógico:**

Con esta reestructuración, la traducción es:

*Entidades fuertes:*

- Clientes(Identificación, Nombre, Dirección, Tipo facturación, Tipo cliente)
- Festivos(Fecha). Es redundante con Tienen, véase más adelante.
- Comunidades(Nombre)

*Entidades débiles:*

- Llamadas(Fecha, Hora, Duración, Clientes.Identificación). Es redundante con Facturan si se incluye el atributo Duración en Facturan, véase más adelante.
- Tarifas(Hora inicio, Hora fin, Tipo tarifa, Tarifa, Comunidades.Nombre)

*Relaciones:*

- Realizan(Clientes.Identificación, Llamadas.Fecha, Llamadas.Hora). Las claves de Clientes y Llamadas deben formar la clave de esta relación porque Llamadas es débil. Esta tabla es redundante con Llamadas.
- Facturan(Clientes.Identificación, Llamadas.Fecha, Llamadas.Hora, Tarifas.Hora inicio, Tarifas.Tipo tarifa, Comunidades.Nombre, N° factura, Mes). La clave de Llamadas es la que debe formar clave de Facturan, debido a su cardinalidad (1,1). Esta tabla contiene toda la información de Llamadas salvo Duración. Se puede omitir la tabla Llamadas añadiendo el atributo Duración a Facturan. Sin embargo, no se puede omitir Tarifas por su restricción de cardinalidad (puede haber tarifas en las que no se hayan realizado llamadas).
- Corresponden(Tarifas.Hora inicio, Tarifas.Tipo tarifa, Comunidades.Nombre). Las claves de Tarifas y Comunidades deben formar la clave de Corresponden porque Tarifas es débil. Esta tabla es redundante con Tarifas. En la simplificación nos quedamos con Corresponden en lugar de con Tarifas para respetar la restricción de cardinalidad apuntada en el punto anterior.



- Tienen(Festivos.Fecha, Comunidades.Nombre). Los dos atributos de Tienen deben formar clave para respetar la restricción de cardinalidad varios a varios. Como el conjunto de entidades Festivos no tiene más atributos que la fecha, se puede eliminar del esquema, ya que la información relevante aparece en Tienen.
- Entre(Comunidades.Nombre origen, Comunidades.Nombre destino, Llamadas.Fecha, Llamadas.Hora, Clientes.Identificación). Se ha renombrado Nombre de Comunidades como Nombre origen y Nombre destino. Como Llamadas tiene restricción de cardinalidad (1,1), su clave puede formar exclusivamente la clave de Entre.

El esquema de la base de datos final queda:

*Entidades fuertes:*

- Clientes(Identificación, Nombre, Dirección, Tipo facturación, Tipo cliente)
- Comunidades(Nombre)

*Relaciones:*

- Realizan(Clientes.Identificación, Llamadas.Fecha, Llamadas.Hora)
- Facturan(Clientes.Identificación, Llamadas.Fecha, Llamadas.Hora, Tarifas.Hora inicio, Tarifas.Tipo tarifa, Duración, Comunidades.Nombre, N° factura, Mes)
- Corresponden(Tarifas.Hora inicio, Tarifas.Tipo tarifa, Comunidades.Nombre)
- Tienen(Festivos.Fecha, Comunidades.Nombre)
- Entre(Comunidades.Nombre origen, Comunidades.Nombre destino, Llamadas.Fecha, Llamadas.Hora, Clientes.Identificación)

### Restricciones de integridad referencial:

Con respecto a Realizan:

- $\Pi_{Identificación}(Re\ alizan) \subseteq \Pi_{Identificación}(Clientes)$
- No hay más con respecto a la otra relación (Llamadas) porque se ha omitido. La tabla Realizan contiene toda la información de Llamadas y se impondrán las restricciones de integridad referencial de otras tablas con respecto a Llamadas con la tabla Realizan.

Con respecto a Facturan:

- $\Pi_{Identificación, Fecha, Hora}(Facturan) \subseteq \Pi_{Identificación, Fecha, Hora}(Re\ alizan)$
- $\Pi_{Hora\ inicio, Tipo\ tarifa, Nombre}(Facturan) \subseteq \Pi_{Hora\ inicio, Tipo\ tarifa, Nombre}(Corresponden)$

Con respecto a Corresponden:

- $\Pi_{Nombre}(Corresponden) \subseteq \Pi_{Nombre}(Comunidades)$
- No hay más con respecto a la otra relación (Tarifas) porque se ha omitido. Las restricciones de integridad referencial con respecto a la tabla Tarifas se impondrán en términos de la tabla Corresponden.

Con respecto a Tienen:

- $\Pi_{Nombre}(Tienen) \subseteq \Pi_{Nombre}(Comunidades)$
- No hay más con respecto a la otra relación (Festivos) porque se ha omitido.

Con respecto a Entre:

- $\Pi_{Nombre\ origen}(Entre) \subseteq \Pi_{Nombre}(Comunidades)$
- $\Pi_{Nombre\ destino}(Entre) \subseteq \Pi_{Nombre}(Comunidades)$
- $\Pi_{Identificación, Fecha, Hora}(Entre) \subseteq \Pi_{Identificación, Fecha, Hora}(Re\ alizan)$

### Restricciones de participación total:

Hay que imponerlas en todas las partes de las relaciones con participación mínima 1.

- Realizan - (1,1) - Llamadas:  
No se impone porque están integradas en una sola tabla (Realizan).
- Llamadas - (1,1) - Facturan:  
 $\Pi_{Identificación, Fecha, Hora}(Facturan) = \Pi_{Identificación, Fecha, Hora}(Re\ alizan)$
- Llamadas - (1,1) - Entre:  
 $\Pi_{Identificación, Fecha, Hora}(Entre) = \Pi_{Identificación, Fecha, Hora}(Re\ alizan)$
- Tarifas - (1,1) - Corresponden:  
No se impone porque están integradas en una sola tabla (Corresponden).
- Festivos - (1,1) - Tienen:  
No se impone porque están integradas en una sola tabla (Tienen).



- 6) (3 puntos) Dada la relación  $R(A,B,C,D)$  y las dependencias funcionales  $AD \rightarrow C$ ,  $C \rightarrow B$  y  $C \rightarrow D$ , se pide:
- Descomponer  $R$  para obtener un conjunto de esquemas en FNBC.
  - ¿La transformación ha preservado las dependencias funcionales?
  - Descomponer  $R$  para obtener un conjunto de esquemas en 3FN.

R:

a) Descomponer  $R$  para obtener un conjunto de esquemas en FNBC.

Hay que comprobar en primer lugar si no está ya en FNBC:

$\{AD\}^+ = \{A,B,C,D\}$  Es superclave

$\{C\}^+ = \{C,B,D\}$  No es superclave

Se descompone  $R$  aplicando el algoritmo de descomposición a FNBC:

$D = \{R\}$

while  $Q \in D$ , t.q.  $D$  no está en FNBC

{encontrar  $X \rightarrow Y$  de  $Q$  que viole FNBC

reemplazar  $Q$  por  $Q-Y$  y  $X \cup Y$ }

$C \rightarrow B$

$Q-Y = S(A,C,D)$

$X \cup Y = T(B,C)$

Con  $T$  se ha terminado porque sólo tiene dos atributos

Para comprobar si  $S$  está en FNBC:

- Determinar el cierre de  $F$  ( $F^+$ ).

- Comprobar que para cada  $X \rightarrow Y \in F^+$ ,  $X$  es superclave.

$\{A\}^+ = \{A\}$

$\{C\}^+ = \{C,D\}$  No es superclave,  $\{C \rightarrow D\}$

$\{D\}^+ = \{D\}$

Se descompone  $S$  con respecto a  $\{C \rightarrow D\}$

$S_1 = \{A,C\}$

$S_2 = \{C,D\}$

Se ha terminado en los dos casos porque son relaciones de dos atributos.

b) ¿La transformación ha preservado las dependencias funcionales?

Para comprobar si preserva las dependencias funcionales se aplica el algoritmo:

for each D.F.  $X \rightarrow Y$

1.  $Z := \{X\}$

while cambios en  $Z$  do

for  $i := 1$  to  $k$  do

2.  $Z := Z \cup ((Z \cap R_i)^+ \cap R_i)$

Si  $Y \subseteq Z \Rightarrow X \rightarrow Y \in G^+$

Si hay algún  $X \rightarrow Y$  t.q.  $Y$  no es subcjto. de  $Z$ , significa que  $X \rightarrow Y$  no  $\in G^+$  y, por tanto, no se conservan las D.F.

Para  $AD \rightarrow C$ :

1.  $Z = \{A,D\}$

2.S1(A,C)  $Z = \{A,D\} \cup ((\{A,D\} \cap \{A,C\})^+ \cap \{A,C\}) = \{A,D\} \cup (\{A\} \cap \{A,C\}) = \{A,D\}$

2.S2(C,D)  $Z = \{A,D\} \cup ((\{A,D\} \cap \{C,D\})^+ \cap \{C,D\}) = \{A,D\} \cup (\{D\} \cap \{C,D\}) = \{A,D\}$

3.T(B,D)  $Z = \{A,D\} \cup ((\{A,D\} \cap \{B,D\})^+ \cap \{B,D\}) =$



$$\{A,D\} \cup (\{D\} \cap \{B,D\}) = \{A,D\}$$

$\{C\}$  no es subconjunto de  $Z$ .  $AD \rightarrow C$  no se preserva, paramos.

c) Descomponer  $R$  para obtener un conjunto de esquemas en 3FN.

En primer lugar se comprueba si se encuentra ya en 3FN:

Para toda  $X \rightarrow Y$ , o bien  $X$  es superclave o  $Y$  contiene algún atributo que pertenece a una clave candidata.

No hay claves candidatas de un atributo.

$$\{AB\}^+ = \{A,B\}$$

$$\{AC\}^+ = \{A,B,C,D\} \text{ Clave candidata}$$

$$\{AD\}^+ = \{A,B,C,D\} \text{ Clave candidata}$$

$$\{BC\}^+ = \{B,C,D\}$$

$$\{BD\}^+ = \{B,D\}$$

$$\{CD\}^+ = \{C,B,D\}$$

Para  $AD \rightarrow C$ ,  $AD$  es superclave

Para  $C \rightarrow B$ ,  $C$  no es superclave y  $B$  no pertenece a ninguna clave candidata. Por lo tanto,  $R$  NO ESTÁ EN 3FN.

Se aplica el algoritmo de descomposición en 3FN:

$D = \{R\}$

1. Encontrar un recubrimiento mínimo  $T$  de  $F$

2. for each  $X \rightarrow Y \in T$ , crear en  $D$  un esquema  $R_i$  con  $\{X \cup \{A_1\} \cup \dots \cup \{A_k\}\}$  si  $R_i \not\subseteq R_j \in D$ , dadas las D.F.  $X \rightarrow \{A_1\}$ , ...,  $X \rightarrow \{A_k\}$

3. Si ninguno de los esquemas contiene una clave de  $R$ , se crea uno nuevo.

1.  $F$  es minimal porque  $C \rightarrow B$  y  $C \rightarrow D$  sólo tienen un atributo en el antecedente y consecuente, y porque de  $AD \rightarrow C$  no se puede eliminar  $A$  o  $D$  sin eliminar su cierre, dado que ni  $A$  ni  $D$  son superclaves y  $AD$  sí.

2.  $AD \rightarrow C$ :  $\{A,C,D\}$

2.  $C \rightarrow B$ :  $\{C,B\}$

2.  $C \rightarrow D$ :  $\{C,D\}$  No se añade porque está incluido en la primera

$R_1(A,C,D)$

$R_2(C,B,D)$