

Reingeniería de Bases de Datos Relacionales: Un Enfoque Basado en el Análisis de Dependencias de Inclusión

Laura C. Rivero ^{*†} Viviana E. Ferraggine ^{*} Jorge H. Doorn ^{*}

Resumen

La conversión de un esquema de bases de datos en otro con mayor contenido semántico es un tema de investigación actual, y un tópico con aplicación en varias áreas de desarrollo tales como integración de esquemas, migración de sistemas “legacy” y reingeniería de modelos de datos desactualizados o de pobre calidad. Las dependencias de inclusión constituyen uno de los conceptos clave para llevar a cabo la reingeniería de esquemas de bases de datos. Con frecuencia, formas atípicas de dependencias de inclusión aparecen como consecuencia de un diseño desnormalizado, por la presencia de construcciones complejas tales como relaciones de orden superior con restricciones de integridad adicionales en forma de relaciones de menor orden, o simplemente cuando objetos de la realidad y sus relaciones han sido mal representados en el modelo conceptual. Específicamente, éste es el caso de reglas del negocio plasmadas incorrectamente como dependencias de inclusión atípicas. Este trabajo presenta un estudio exhaustivo de las dependencias de inclusión y un análisis de su posible origen, en el contexto de la reingeniería de esquemas relacionales. Se proponen heurísticas para la conversión de esquemas conteniendo tales dependencias en otros equivalentes con sólo restricciones de integridad referencial. En caso de encontrar dependencias atípicas remanentes, se provee una interpretación de su significado y necesidad de mantenimiento.

Palabras Clave: *Modelo Entidad-relación, dependencia de inclusión, diseño de bases de datos, reingeniería de esquemas conceptuales, modelo de datos semántico.*

1 Introducción

En muchas organizaciones hay bases de datos que han evolucionado con el correr del tiempo. En estos sistemas, la comprensión de los datos se ha perdido parcial o totalmente o bien ya no refleja el UdeD (Universo del Discurso) actual, restringiendo por esta razón el uso efectivo que la organización puede hacer de su información. En otras circunstancias algunas dependencias han sido detectadas o previstas en estadios tempranos del diseño pero luego, por alguna razón, fueron incorrectamente representadas. Esquemas de baja calidad pueden ser producidos cuando las tareas de diseño se realizan desviándose de las bien conocidas heurísticas que guían la formalización de aplicaciones y la optimización en tiempo de diseño [2] o cuando la captura de requisitos desde el UdeD no ha sido apropiadamente validada.

* INTIA – Departamento de Computación y Sistemas - Universidad Nacional del Centro. Campus Universitario, 7000, Tandil, Buenos Aires, ARGENTINA

† LINTI – Universidad Nacional de La Plata, 1900, La Plata, ARGENTINA

Esquemas inicialmente correctos pueden deteriorarse con el tiempo debido a un mantenimiento inadecuado. En estos casos, la reingeniería de los sistemas de software es crucial. El objetivo es recomenzar con una implementación más clara que facilite el futuro mantenimiento y evolución. Herramientas de software, guías, heurísticas, etc. podrían marcar una importante diferencia en el proceso de mejora de la calidad de la base de datos.

El diseño de esquemas relacionales partiendo de un modelo semántico genera restricciones de integridad referencial típicas. Estas restricciones y las acciones referenciales correspondientes encierran la semántica local y global que es inducida naturalmente por el UdeD, y que el diseñador tiene en mente durante el proceso de desarrollo. En otro escenario, la descomposición y la síntesis que surgen del proceso de normalización/desnormalización puede no tener en cuenta esas restricciones. Un refinamiento ad-hoc del diseño lógico sin tomar en cuenta el correspondiente diseño conceptual, usualmente resulta en algún nivel de desnormalización o en la presencia de construcciones n-arias complejas que pueden resultar en el modelado de restricciones referenciales atípicas.

Errores de diseño de esquemas de bases de datos reales frecuentemente toman la forma de dependencias de inclusión no basadas en claves, revelando la presencia de entidades ocultas e interrelaciones incomprensibles. La transformación de construcciones n-arias en esquemas de bases de datos puede generar algunas estructuras de dependencias de inclusión inusuales, con el propósito de mantener, no sólo la equivalencia estructural, sino también la semántica.

El diseñador podrá comprobar, a través de una profunda inspección del UdeD, que es muy difícil, si no imposible, encontrar vínculos semánticos que justifiquen tales dependencias no convencionales, en el mundo de la base de datos. Sin embargo, del estudio de varios casos reales surge que estas construcciones se encuentran frecuentemente. Todos los posibles factores de distorsión, sospechosos de causar estos tipos de dependencias, deberían ser considerados cuando se realiza un rastreo reverso, desde el modelo conceptual al UdeD. A través de esta regresión, el diseñador estará habilitado para descubrir no sólo las nociones embebidas en la dependencia (con el objetivo de especificarlas correctamente) sino también el proceso que justifica su generación.

Como puede leerse en [9], "...Integridad referencial es una aplicación particular de una dependencia de inclusión. Tal restricción requiere que el conjunto de distintos valores de alguna columna especificada, simple o compuesta, debe ser un subconjunto de los valores de otra columna especificada (simple o compuesta, respectivamente). En el caso de restricciones de integridad referencial (*rir*, dependencias de inclusión basadas en claves), el conjunto de distintos valores de clave extranjera debería ser un subconjunto del conjunto de distintos valores de clave primaria, tomados del mismo dominio. Las dependencias de inclusión puras (*di*), sin embargo, pueden ser aplicadas entre otros pares de atributos (es decir, no claves) ...". Las *rir* pueden especificarse declarativamente (FOREIGN KEY ... REFERENCES ...) y ser forzadas por la mayoría de los sistemas relacionales y posrelacionales actuales, mientras que las *di* se definen usualmente a través de cláusulas CHECK o triggers. Dado que estas últimas no pueden ser definidas en la misma forma que las *rirs*, su presencia frecuentemente provoca un esfuerzo extra en el desarrollo de programas de aplicación y el mantenimiento de integridad.

Este trabajo está destinado a dar un marco de referencia para el estudio de *dis* y *rirs*, en principio desde un punto de vista sintáctico. Luego se desarrolla un conjunto de hipótesis que sustentan la presencia de *dis* y *rirs* atípicas, analizando su posible origen desde una perspectiva semántica. Se describe una heurística que permite la conversión de un esquema relacional en uno normalizado, a través de la detección y correcta especificación de entidades y relaciones existentes en el UdeD pero no representadas explícitamente en el esquema, revelando en este proceso, algunas reglas del negocio indefinidas. Las *dis* se convierten en *rirs* hasta que se alcanza un punto de irreducibilidad. Para concluir, se establecen las condiciones de contexto que indican cuándo las *dis remanentes (dirs)* deben ignorarse o mantenerse.

Este artículo ha sido organizado como sigue: la Sección 2 describe la terminología y notación utilizada. La Sección 3 presenta un análisis completo de las *rirs*, mientras que la Sección 4 está dedicada al estudio de *dis*. La Sección 5 contiene un enfoque heurístico para convertir esquemas conceptuales, incluyendo la materialización de las *dirs*. Las dos últimas secciones están dedicadas a describir brevemente los trabajos relacionados y las conclusiones y trabajos en marcha.

2 Terminología y Notación

2.1 Esquemas relacionales

Un esquema relacional se indica $\mathbf{R}=\langle\mathbf{R},\mathbf{D}\rangle$. $\mathbf{R}=\{R_1, R_2, \dots, R_q\}$ y $\mathbf{D}=\{\mathbf{DF}, \mathbf{DI}\}$ representan el conjunto de esquemas de relaciones y el conjunto de restricciones respectivamente. \mathbf{DF} y \mathbf{DI} son los conjuntos de dependencias funcionales y de inclusión. El esquema de cada relación R_j ($j=1, 2, \dots, q$) tiene un nombre, un conjunto de atributos, una *clave primaria* y tal vez una o más claves alternativas. La clave primaria se denotará \mathbf{K}_j . El conjunto de atributos se denota $\mathbf{atr}(\mathbf{R}_j)$.

Una *dependencia funcional (df)* en R_j se expresa: $\mathbf{R}_j:\mathbf{X}\rightarrow\mathbf{Y}$. Esto significa que para todo par de tuplas t, t' en R_j , si $t[\mathbf{X}]=t'[\mathbf{X}]$, entonces $t[\mathbf{Y}]=t'[\mathbf{Y}]$. Si \mathbf{X} es la clave de R_j , la dependencia funcional es *primaria*. De otra forma resulta *secundaria*. Cuando del contexto de la discusión quede claro cuál es R_j , se omitirá de la expresión.

Formalmente, una *dependencia de inclusión (di)* es una expresión $\mathbf{R}_i[\mathbf{W}]\subseteq\mathbf{R}_d[\mathbf{Y}]$. R_i y R_d son nombres de relaciones (posiblemente el mismo); $R_i[\mathbf{W}]$ y $R_d[\mathbf{Y}]$ se denominan términos izquierdo y derecho de la dependencia, respectivamente. \mathbf{W} y \mathbf{Y} son conjuntos de atributos compatibles, es decir $|\mathbf{W}|=|\mathbf{Y}|$ y el n -ésimo atributo de R_i pertenece al mismo dominio de definición que el n -ésimo de R_d . R_i y R_d satisfacen esta restricción si y sólo si para cada tupla t_i de R_i , existe al menos una tupla t_d en R_d tal que $t_i[\mathbf{W}]=t_d[\mathbf{Y}]$. La Figura 1 ilustra la notación gráfica utilizada en este trabajo para *dis*

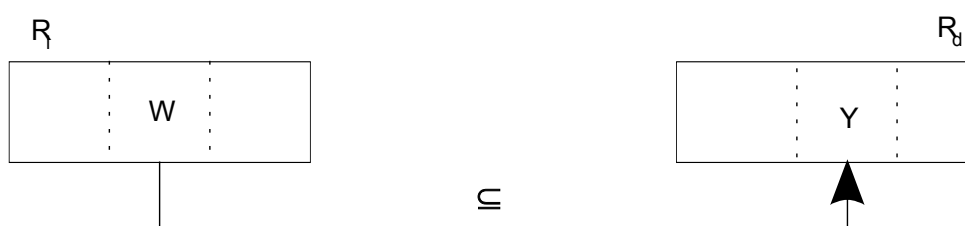


Figura 1. Notación Gráfica para dependencias de Inclusión

Cuando \mathbf{Y} es la clave primaria de R_d (\mathbf{K}_d), la *di* es basada-en-clave y habitualmente se denomina *restricción de integridad referencial (rir)* o simplemente *referencia*. En este caso \mathbf{W} constituye una *clave extranjera* para R_i (\mathbf{FK}_i). La *rir* se establece como $\mathbf{R}_i[\mathbf{FK}_i]\subseteq\mathbf{R}_d[\mathbf{K}_d]$ [1, 11, 15].

2.2 Entidades y claves

En el mundo real, las cosas tienen propiedades. Naturalmente, no hay dos cosas que tengan exactamente las mismas propiedades, pues al menos diferirán en su ubicación física. Este principio implica que las claves serían innecesarias a los efectos de individualizar un objeto del mundo real [31].

Cuando los humanos necesitan identificar una determinada cosa usan una o más propiedades que caracterizan esas cosas, si esto les resulta práctico. Si no resulta práctico, agregan rótulos artificiales a las cosas, para facilitar su identificación. Los primeros identificadores, denominados *claves inteligentes* [10] cubren dos roles: tienen significado y sirven como identificadores. Los segundos se denominan *claves surrogantes* y sólo cumplen el rol de identificadores.

En otras palabras, las *claves surrogantes* preceden el procesamiento computacional de los datos. Estas claves deben ser memorizadas, por lo tanto deberían ser el último recurso. Sin embargo, su uso continuado las ha convertido en atributos regulares. Ejemplos de esta práctica son los nombres de las personas en la antigüedad, luego los apellidos y actualmente, por razones administrativas generalmente, el número de DNI[‡] o el número de socio de un club. En el mundo de las bases de datos, las claves son conceptos relevantes, que tienen dos propiedades: unicidad e irreducibilidad [1, 9, 10, 15, 30].

La propiedad de unicidad merece un análisis más profundo. Una clave dada, surrogante o inteligente, es usualmente elegida para servir en un determinado contexto o alcance. Por ejemplo, el DNI tiene como alcance el territorio de la República Argentina, el número de socio de un club vale en el club, etc. Siempre que la necesidad de identificación se conserve dentro de este contexto, la clave es útil; de otra forma, la clave debe ser reemplazada por una nueva que tenga la propiedad de unicidad en un ámbito más amplio. Habitualmente, este reemplazo no es “realista” y pierde el conocimiento acerca del ámbito original al que pertenece. En la práctica, esto se resuelve extendiendo la clave original con el identificador del ámbito. Dado que en la mayoría de los casos el ámbito es una cosa, lo que realmente ocurre es que la nueva clave se apropia del identificador del contexto. En la terminología de base de datos, estas entidades se denominan “débiles” y el identificador original se convierte en el “fragmento débil” o “clave parcial” del nuevo identificador [10, 15].

Que un esquema se encuentre al menos en 3FN garantiza que cada relación se corresponde con un solo tipo de noción del dominio de la aplicación (objeto o relación). Las excepciones a esta regla están dadas por tablas con más de una clave o aquéllas que fusionan tipos con subtipos. Por otra parte, una relación con más de una clave, no necesariamente representa más de un objeto del UdeD, sino por ejemplo, varios roles del mismo. Estas cuestiones semánticas deben ser elucidadas por el usuario. Como se verá en las Secciones 3.2, 3.3 y 4., desde una base puramente sintáctica no es posible distinguir los diferentes casos.

3 Análisis de las *rirs*

La comprensión de las características semánticas referidas a las nociones de *rirs* y *dis* es facilitada por el análisis de los atributos involucrados. Un criterio que resulta totalmente independiente del dominio de aplicación consiste en el estudio de los términos derecho e izquierdo de la dependencia, considerando sus diferentes ubicaciones en relación con la disposición de la clave primaria[§]. Siendo W un conjunto de atributos de R , K la clave primaria de R y Z un subconjunto de atributos secundarios, los cinco tipos o correlaciones posibles se ilustran en la Figura 2: I) $W \equiv K$; II) $W \equiv Z$; III) $W \equiv K^1$, siendo K^1 un subconjunto estricto de K , $K^1 \neq \emptyset$; IV) $W \equiv K \cup Z$; y finalmente V) $W \equiv K^1 \cup Z$, $K^1 \neq \emptyset$. En todos los casos $Z \neq \emptyset$.

[‡] DNI: Documento Nacional de Identidad, en Argentina.

[§] Este enfoque se ha denominado también correlación entre claves [29]. Los autores sólo distinguen 4 casos.

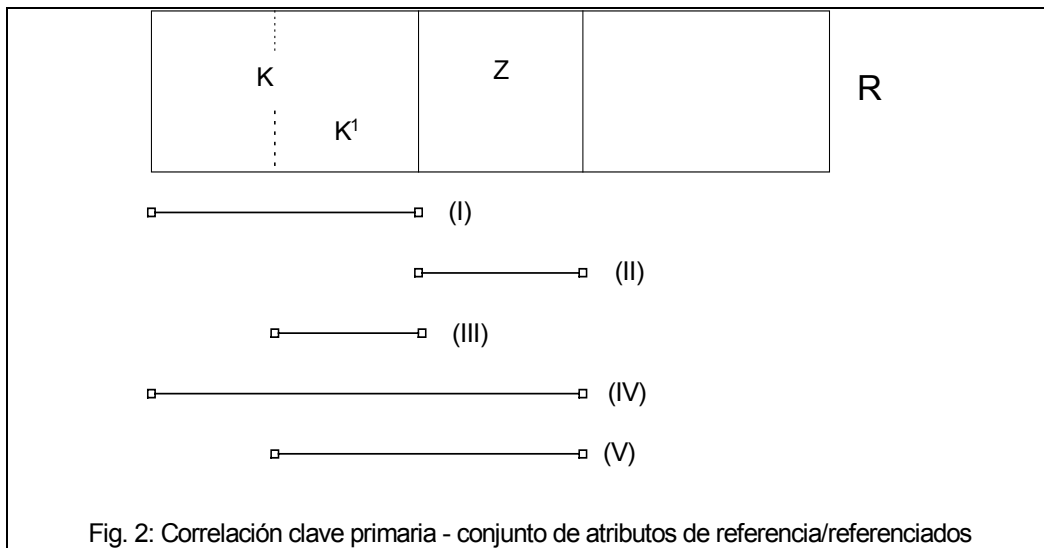


Fig. 2: Correlación clave primaria - conjunto de atributos de referencia/referenciados

Teniendo en cuenta estas disposiciones, pueden obtenerse 25 casos posibles de pares $\langle R_i[W_i], R_d[W_d] \rangle^{**}$. Aquéllos casos numerados de 1 a 5 en la Tabla 1 corresponden a *rirs* aunque algunos casos ocultan tipos especiales de *dis*. *Rirs* de tipos IV y V, es decir con el término izquierdo de tipo IV o V, son atípicos y habitualmente aparecen como consecuencia de cambios informales en el diseño lógico [14]. Los casos remanentes representan *dis*. Nótese que, en el contexto del análisis de las restricciones de integridad referencial, la estructura del término izquierdo determina su tipo. Sin embargo, en el caso de las restricciones de tipos IV y V, también la estructura del término derecho debe ser analizada. El estudio de las *dis* debe enfocarse principalmente sobre el término derecho. El resto de esta sección se dedica al análisis de *rirs*.

Estas restricciones son fundamentales ya que resultan suficientes para modelar, de manera natural, las relaciones entre entidades que habitualmente capturan los modelos semánticos [1]. *Rirs* de tipos I, II y III constituyen la representación de tales relaciones. Por ejemplo, el tipo I permiten la representación de relaciones de tipo-subtipo tales como “cada vendedor es una persona”. Las de tipo II permite abstraer relaciones designativas de cardinalidad 1:1, N:1 o n-arias con al menos una cardinalidad 1. Finalmente, las de tipo III aparecen en relaciones asociativas N:N, n-arias en general y las correspondientes a los vínculos entidad fuerte-entidad débil [10, 15]. Los casos IV y V merecen un análisis más detallado. Las restricciones de la primera columna se denominarán *rirs* de tipo I, II, II, IV y V respectivamente, de acuerdo a la denominación del término izquierdo (según la notación de la Figura 2).

En las siguientes secciones el análisis se centra en entidades y relaciones (tablas o relaciones en el mundo de la base de datos), cuyas claves están compuestas de dos o más atributos. En el análisis del lado derecho de *rirs* no convencionales, desarrollado en las Secciones 3.2 y 3.3 se excluyen aquellas entidades regulares que tienen dos o más atributos en su identificador, y que naturalmente no tienen significado en el UdeD si se los considera en forma aislada. Teniendo en cuenta que no pueden establecerse sobre ellos otras dependencias funcionales más que las triviales, a los efectos de identificar tales entidades estos conjuntos de atributos deben ser considerados una unidad indivisible.

** Subíndices ‘i’ y ‘d’ representan ‘izquierdo’ y ‘derecho’ respectivamente.

W_i	W_d	I) Clave (K_d)	II) No Clave (Z_d)	III) Parte de Clave (K_d^1)	IV) Clave + No Clave ($K_d \cup Z_d$)	V) Parte de Clave + No Clave ($K_d^1 \cup Z_d$)
I) Clave (K_i)		1. $K_i \ll K_d$	6. $K_i \subseteq Z_d$	11. $K_i \subseteq K_d^1$	16. $K_i \subseteq K_d \cup Z_d$	21. $K_i \subseteq K_d^1 \cup Z_d$
II) No Clave (Z_i)		2. $Z_i \ll K_d$	7. $Z_i \subseteq Z_d$	12. $Z_i \subseteq K_d^1$	17. $Z_i \subseteq K_d \cup Z_d$	22. $Z_i \subseteq K_d^1 \cup Z_d$
III) Parte de Clave (K_i^1)		3. $K_i^1 \ll K_d$	8. $K_i^1 \subseteq Z_d$	13. $K_i^1 \subseteq K_d^1$	18. $K_i^1 \subseteq K_d \cup Z_d$	23. $K_i^1 \subseteq K_d^1 \cup Z_d$
IV) Clave + No Clave ($K_i \cup Z_i$)		4. $K_i \cup Z_i \ll K_d$	9. $K_i \cup Z_i \subseteq Z_d$	14. $K_i \cup Z_i \subseteq K_d^1$	19. $K_i \cup Z_i \subseteq K_d \cup Z_d$	14. $K_i \cup Z_i \subseteq K_d^1 \cup Z_d$
V) Parte de Clave + No Clave ($K_i^1 \cup Z_i$)		5. $K_i^1 \cup Z_i \ll K_d$	10. $K_i^1 \cup Z_i \subseteq Z_d$	15. $K_i^1 \cup Z_i \subseteq K_d^1$	20. $K_i^1 \cup Z_i \subseteq K_d \cup Z_d$	25. $K_i^1 \cup Z_i \subseteq K_d^1 \cup Z_d$

Tabla 1. Tipos de dependencias de inclusión

Referencias: K_* (Clave); Z_* (atributos no-clave o secundarios); K_*^1 (subconjunto estricto de K_*). $*$ = i, d

3.2 Relaciones izquierda y derecha para *rirs* de tipo IV: $R_i[K_i \cup Z_i] \ll R_d[K_d]$ (Celda 4 en Tabla 1).

Para los análisis de casos de esta sección y de las siguientes se han bosquejado ejemplos que pueden resultar exagerados para un diseñador con cierta experiencia. Sin embargo se incluyen por razones explicativas, reiterando que muchas de estas situaciones atípicas fueron detectadas en sistemas reales, en la mayoría de los casos por haber plasmado erróneamente reglas del negocio sin tener en cuenta buenas prácticas de diseño, heurísticas convenientes, etc.

Sea el siguiente esquema relacional con relaciones en FNBC (Forma Normal de Boyce-Codd). Las claves se subrayan.

(R_i) INSCRIPTO-EN-CONCURSO (#Estudiante, #Curso, FechaInscripción) y
 (R_d) ESTUDIANTE-APROBADO (#Estudiante, #Curso, NotaFinal).

R_i representa la nómina de estudiantes registrados para concursar por un cargo de auxiliar en un curso y R_d es la lista de notas finales de los alumnos en los cursos que han aprobado. Las dependencias funcionales en este esquema son:

DF = { R_i : #Estudiante \rightarrow {#Curso, FechaInscripción}; R_d : {#Estudiante, #Curso} \rightarrow NotaFinal}

Y el conjunto de *dis* y *rirs* **DI** = { R_i [#Estudiante, #Curso] \ll R_d [#Estudiante, #Curso]}

Nótese que este caso ha sido mal modelado, pues el atributo #Curso tiene diferentes semánticas en R_i y R_d , contradiciendo de esta forma el Principio de la Relación Universal. Así, los pares #Estudiante, #Curso en R_i y R_d tienen diferentes significados. En general, si en las tablas derecha e izquierda se establecen diferentes dependencias funcionales involucrando los mismos atributos, puede concluirse que existen algunos problemas de semántica, debidos a la sobrecarga de uno o más nombres de atributos que representan diferentes significados. En el ejemplo, una manera de interpretar la *rirs* es una *regla del negocio*: “Los estudiantes pueden

inscribirse en concurso para desempeñarse como auxiliares, sólo si han aprobado el examen de ese curso". Una forma más apropiada de representar este concepto del UdeD proviene del análisis de la estructura de las tablas derecha e izquierda. La tabla derecha es una relación que podría ser derivada de una relación N:N entre ESTUDIANTE y CURSO. Un Diagrama Entidad Relación (DER) ilustrativo de este caso podría ser el de la Figura 3. R_2 es la relación designativa que produce la *rir* bajo análisis.

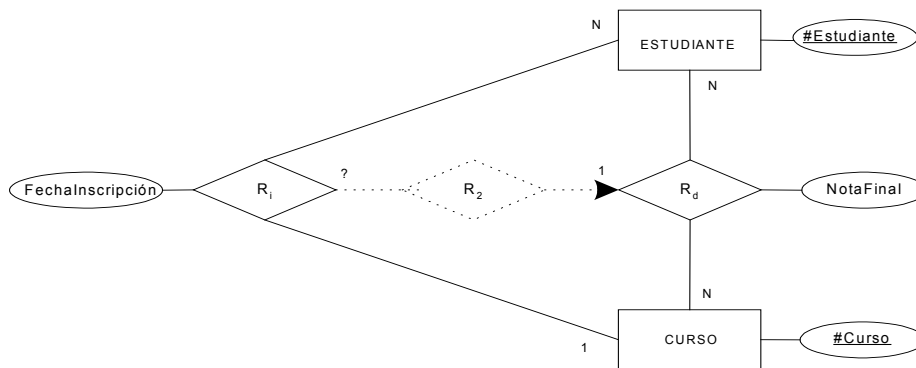


Figura 3. Un DER posible para el ejemplo

Las expresiones en cursiva resaltadas entre comillas en esta sección y en las que siguen, son un caso particular de *reglas del negocio* denominadas *restricciones de dominio*. Estas restricciones controlan los valores de un atributo respecto de los elementos de un conjunto de valores válidos.

Debe puntualizarse que este esquema está basado en una extensión del Modelo de Entidades y Relaciones (MER) [8] que permite agregaciones, es decir relaciones no sólo entre entidades sino también entre entidades y relaciones y entre dos o más relaciones. En segundo lugar, la semántica de la aplicación no queda expresada ya que R_2 (intencionalmente dibujada en líneas punteadas) al ser derivada en el esquema relacional, debería basarse en la clave de R_1 .

Desde otra perspectiva, observando el dominio de la aplicación, el análisis podría ser el siguiente: #Curso en R_1 y R_d , ha sido considerado en diferentes contextos. Lo mismo ocurre con #Estudiante. La suposición de existencia de atributos inaplicables [9] ayuda a detectar una relación de tipo-subtipo (Figura 4). La flecha punteada ilustra la *regla del negocio*.

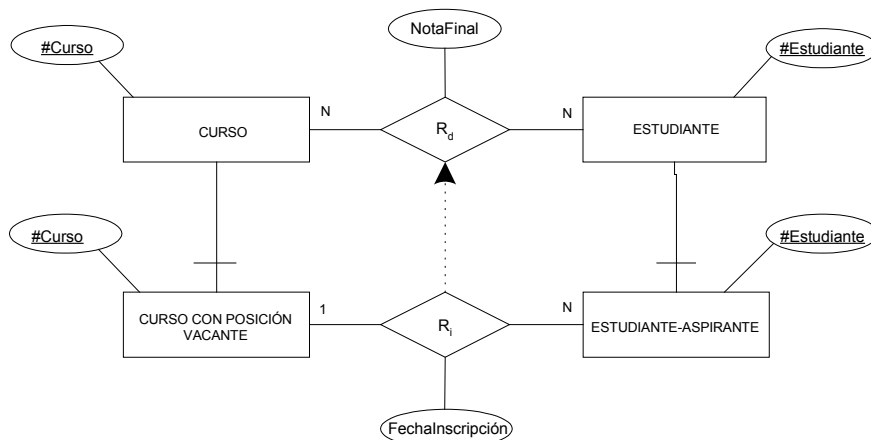


Figura 4. Otro DER posible para el ejemplo

Para analizar el tipo IV desde una perspectiva general, considérese el esquema de la Figura 5. El análisis puede realizarse a través del estudio del término derecho, centralizándose en las diferentes estructuras y su posible origen y significado. Ya que el término derecho tiene al menos dos atributos, se estudiarán todas las posibles configuraciones que tienen claves de dos o más dominios. Se han detectado los siguientes casos:

a) R_d es una relación binaria asociativa $N:N$

Considere las correspondencias siguientes entre atributos $K_i \equiv K_d^1$ y $Z_i \equiv K_d^2$. Como se ha indicado anteriormente, las dependencias funcionales conllevan diferente semántica. Si Z_i es sólo un atributo secundario de R_i , puede concluirse que su coincidencia con K_d^2 es debida sólo a la coincidencia de nombres, no representando ninguna noción identificable en el UdeD. Así, se asume que Z_i es también una clave extranjera. Este caso puede representarse como en la Figura 6.

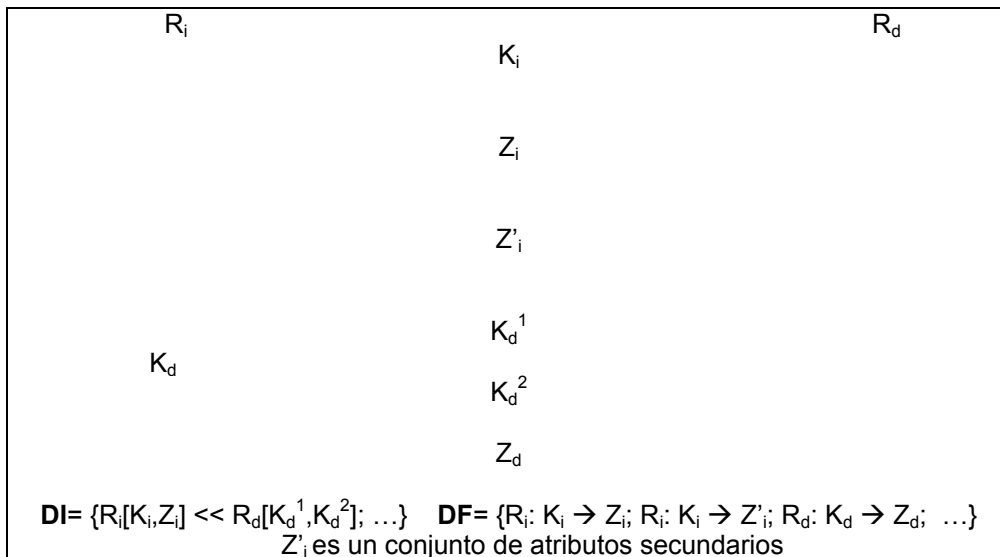


Figura 5. Rir de tipo IV

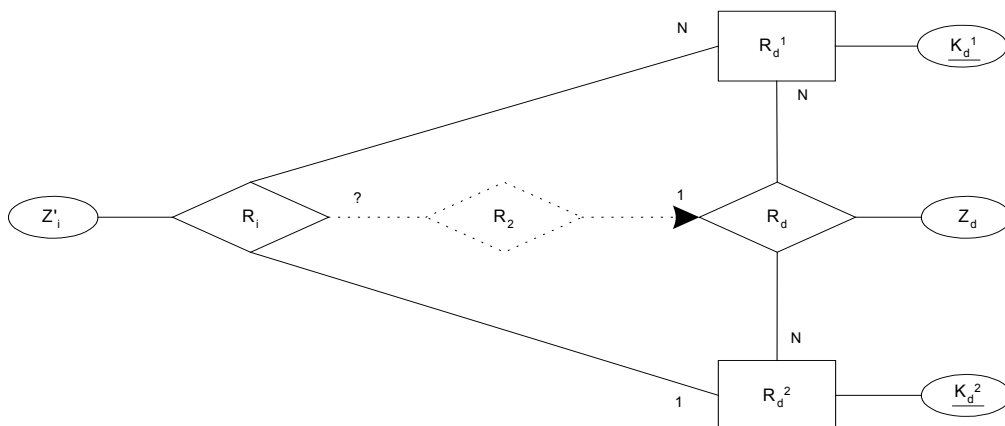


Figura 6. Un DER posible para el caso 3

R_2 indica la *regla del negocio* que representa la inclusión en **DI**.

Un análisis más detallado de R_d , puede mostrar atributos inaplicables incluidos en una o ambas entidades en la relación N:N. En este caso, un DER más apropiado es ilustrado en la Figura 7.

La *regla del negocio* (flecha punteada) podría expresarse:

CHECK ((K_i, Z_i) MATCH UNIQUE (SELECT K_d^1, K_d^2 FROM R_d))
 (1)

A través de este análisis se ha supuesto, sin pérdida de generalidad, que $K_i \equiv K_d^1$ y $Z_i \equiv K_d^2$, siendo K_i y/o Z_i compuesto por uno o más atributos. Si $K_i Z_i$ (y obviamente K_d) tiene más de dos atributos, teniendo en cuenta que $K_i Z_i \equiv K_d$, podría estar ilustrando otros casos (raros).

Supóngase que R_d es una relación binaria asociativa de R_d^1 (con clave K_d^1) y R_d^2 (con clave K_d^2), con K_d^1 (o K_d^2 o ambos) teniendo dos o más atributos. Los componentes de $K_i Z_i$ y K_d podrían no concordar (Figura 8).

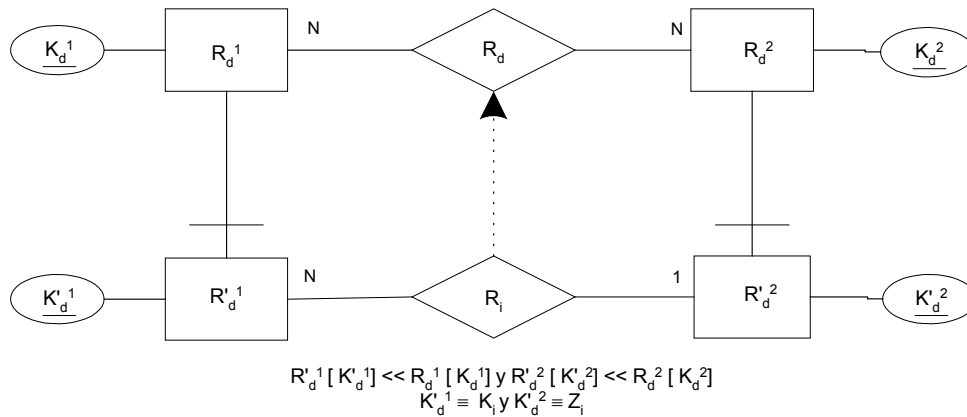


Figura 7. DER semánticamente correcto del caso a)

K_i
Z_i
a_1
...
a_{m1}
a_{m1+1}
...
a_{m2}
a_{m2+1}
...
a_m
b_1
...
b_{m1}
b_{m1+1}
...
b_{m2}

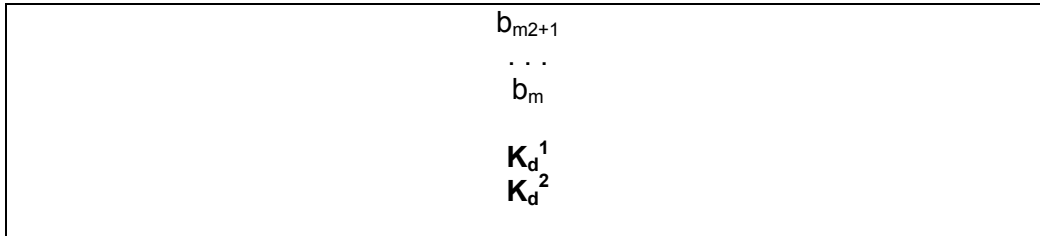


Figura 8. Correspondencia de atributos

En este caso, no se cumple $K_i \equiv K_d^1$ ni $Z_i \equiv K_d^2$ (simétricamente, $Z_i \not\equiv K_d^1$ y $K_i \not\equiv K_d^2$). Si K_i y/o Z_i no son atómicos, se requiere un análisis recursivo de cada parte. Para simplificar este análisis, considérese que K_d tiene tres atributos, A, B y C. Teniendo en cuenta las conformaciones de los términos izquierdo y derecho de la *rir*, podría ocurrir que A, B y C sean identificadores de sus respectivas entidades. En este caso R_d es una relación ternaria N:N:N y el análisis es una generalización del desarrollado en esta sección, pero basado en relaciones ternarias. Otra situación se da cuando uno o dos de estos atributos son fragmentos débiles del identificador de una entidad débil. Dado que estos atributos carecen de significado fuera del ámbito de la entidad débil, algunos de estos casos son incorrectos.

En resumen, el significado preciso de las construcciones del modelado conceptual requiere reducir la ambigüedad semántica evitando la sobrecarga de atributos. Los supuestos de la Relación Universal y del Rol Único constituyen el marco conceptual para el análisis de estos casos, que no serán considerados en el resto del trabajo.

b) *R_d es una entidad débil*

El fragmento de la clave que representa la entidad fuerte o regular (sin pérdida de generalidad K_d^1) puede coincidir con la clave de R_i ($K_i \equiv K_d^1$) o con el atributo o conjunto de ellos Z_i ($Z_i \equiv K_d^1$).

El primer caso puede analizarse sintácticamente en forma similar al caso de la Subsección a), mientras que la última opción parece carente de sentido dado que, cuál sería el significado real de la coincidencia entre el identificador de R_i con el fragmento débil de la clave de R_d ? El rastreo de este concepto en el UdeD seguramente pondrá de manifiesto el mal uso de un atributo.

A modo de ejemplo, considérese el siguiente esquema relacional (claves subrayadas):

PAÍS (IdPaís, NombrePaís, Capital, Población) (regular)
 CIUDAD (NombreCiudad, País, Intendente) (débil)

Las *rirs* son:

CIUDAD[País] << PAÍS[IdPaís] y País[IdPaís, Capital] << CIUDAD[NombreCiudad, País].

La última *rir* es de tipo IV pero podría ser establecida como una de tipo II. La ciudad Capital en un PAÍS podría ser especificada como una clave extranjera mediante la duplicación del atributo IdPaís, de la siguiente forma:

PAÍS (IdPaís, NombrePaís, IdPaís2, Capital, Población) con la *rir* PAÍS[IdPaís2, Capital]<<CIUDAD[NombreCiudad, País]. Sin embargo, esta forma de especificación requiere una restricción adicional, ya que IdPaís debe ser igual a IdPaís2, para cada PAÍS.

En los casos analizados en esta sección, como también en los que se desarrollarán en las Subsecciones c) y d) a continuación, debe hacerse una aclaración respecto de una clase especial de atributos, que aparecen como parte de los identificadores de entidades débiles. Por ejemplo, entidades “sólo clave” son designadas frecuentemente como atributos. Sin embargo, al menos en etapas tempranas del diseño estas entidades deberían ser preservadas y especificadas a través de sus claves. Fechas, órdenes, prioridades, grados de parentesco, etc. son candidatos a ser mal utilizados como atributos, en lugar de entidades. La falta de rigor en esta representación produce la aparición de dependencias funcionales ocultas, que generalmente pasan desapercibidas y quedan sin ser modeladas [14].

c) R_d es una relación ternaria o de mayor orden

El análisis de este caso es una mera extensión del desarrollado en la Subsección a) (Fig. 9). Al igual que en ese caso, R_2 ha sido graficado en líneas punteadas, puesto que no es correcta de acuerdo a las prescripciones para la conversión MER \rightarrow tablas relacionales. Una vez más, la presencia de atributos inaplicables auxilia en el descubrimiento de *reglas del negocio* ocultas.

Algunas relaciones ternarias o de mayor orden pueden ser convertidas en un conjunto de relaciones binarias utilizando un gerundio o relación intersección, o utilizando relaciones de agregación [3, 4, 12, 13, 15, 19, 20, 21, 24]. La representación de la relación de orden mayor que 2 a través de una relación intersección, a pesar de ser artificial, facilita el análisis sintáctico de R_d . Las consideraciones realizadas con respecto a los fragmentos que componen el término izquierdo en la Subsección a) pueden aplicarse en el análisis de este caso.

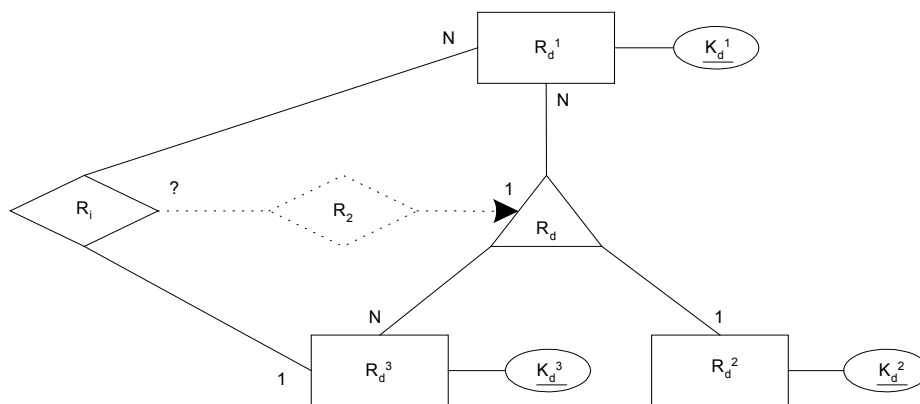


Figura 9. Un DER posible para el caso c)

d) R_d es una relación desnormalizada que podría ser proyectada a 4FN ó 5FN.

Este caso es muy específico, y no se incluirá un análisis detallado del mismo. Debe notarse que, dejando de lado los casos triviales, se asume que R_d tiene un identificador de al menos tres atributos. Que no esté en la máxima forma normal significa que se ha planteado como una relación ternaria (al menos) cuando un análisis algo más profundo del caso hubiese permitido modelar esta relación como un conjunto de relaciones binarias R_d^1, R_d^2, \dots , en 4FN ó 5FN según corresponda [ver refs. 10, 15, 30]. Estas proyecciones deben ser estudiadas para constituir el término derecho de las *rirs* que reemplacen la referencia $R_i [A,B,C] \ll R_d [A,B,C]$, inicialmente planteada.

3.3 Relaciones derecha e izquierda para *rirs* de tipo V: $R_i[K_i^1 \cup Z_i] \ll R_d[K_d]$ (celda 5 en Tabla 1)

La Figura 10 ilustra el esquema de este caso. Nuevamente el análisis debe centrarse en los fragmentos que componen ambos términos de la restricción. Las consideraciones realizadas respecto del término derecho en la sección anterior, son aplicables también en este caso.

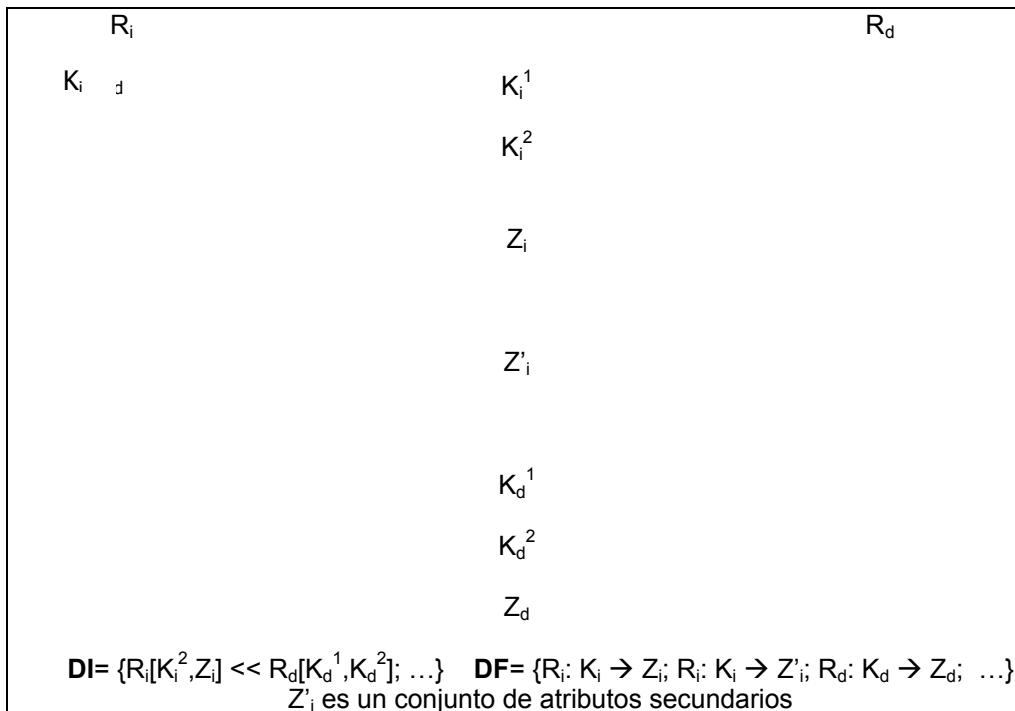


Figura 10. Rir de tipo V

Respecto del izquierdo, los casos de interés detectados son:

a) K_i² Z_i es una clave alternativa de R_i.

a1) En este caso el diseño no sería incorrecto. Por ejemplo, si R_i es una relación ternaria de cardinalidad 1:1:1 ó N:1:1 [7, 18].

Un caso particular: K_i¹K_i² ↔ K_i²Z_i conforman un diccionario (ni K_i¹ → Z_i ni Z_i → K_i¹ se satisfacen). Considérese el siguiente ejemplo: sean las relaciones

LOTE_ESCRITURADO (Calle, Número, CódigoPostal, IdManzana, IdLote, Fechaescritura) con clave alternativa Calle, Número, CódigoPostal.

LOTE (CódigoPostal, Idmanzana, IdLote, Propietario, Superficie)

En este caso, la relación LOTE_ESCRITURADO sería subtipo de LOTE. Nótese que IdManzana, IdLote e inclusive el domicilio dado por Calle y Número son atributos que difícilmente puedan manifestarse en el UdeD como identificadores de sus respectivas entidades. En los dos primeros casos porque simplemente son enumeraciones de partes y en el

segundo porque la clase de entidades correspondientes sería el conjunto de todos los domicilios posibles, lo que no tiene sentido fuera del contexto de una localidad, indicada en este caso por el CódigoPostal. Este análisis es análogo al realizado en la Subsección d) de la Sección 3.2. respecto del tipo IV.

a2) $Z_i \rightarrow K_i^2$, configurando de esta forma un caso típico de relación que no se encuentra en FNBC y cuya normalización conduciría a proyecciones dependientes [10, 30].

b) $K_i^2 Z_i$ no es una clave alternativa de R_i

En este caso valen consideraciones similares a las efectuadas para el tipo IV. Los escenarios posibles son:

b1) K_i^2 puede ser el fragmento débil del identificador de una entidad débil (incorrecto).

b2) Z_i podría ser el fragmento débil, conformando junto con K_i^2 la clave de otra entidad (no de R_i como en el caso a). Este caso también es incorrecto pues debería ser especificado como clave extranjera y Z_i debe incluirse en K_i .

b3) K_i^2 y Z_i son claves de sendas entidades, por ejemplo si R_i es una relación N:N:1 (respectivamente, K_i^1 , K_i^2 , Z_i)

b4) K_i^2 es el fragmento fuerte de la clave de R_i , que en este caso es débil.

Al igual que en caso IV, el análisis de las diferentes combinaciones de los términos puede ser realizado desde un punto de vista sintáctico, pero siempre con referencia a los aspectos del UdeD que pudieron llevar al diseñador a establecer la restricción de tipo V. Z_i no es un atributo común, sino la clave de una entidad, ya esté formalmente expresada u oculta. Aún sin avanzar sobre los aspectos semánticos, algunas combinaciones pueden ser descartadas, bajo la hipótesis de cierto grado de experiencia de quien diseña. Por ejemplo, si K_i^2 es el fragmento débil del identificador de una entidad débil (b1), acompañado de Z_i parecería conformar el identificador de una entidad fuerte pero, tiene sentido la combinación de estas partes?. Todo parece indicar que el fragmento débil ha sido sobrecargado con diferentes semánticas. No ocurre lo mismo si se trata del fragmento fuerte, tal como se puede observarse en el ejemplo de la Subsección a) de esta sección.

A partir del análisis combinado de R_i y R_d , nótese que el MER (o bien alguna de sus extensiones) no es suficiente para efectuar un rediseño del esquema conceptual.

3.4 Comentarios finales sobre las *rirs* de tipos IV y V

Para completar el análisis de estos casos, se efectúan algunos comentarios acerca de la procedencia del término izquierdo. Ya se ha mencionado que los tipos I, II y III provienen de un diseño normalizado, y que por esta razón los vínculos que representan se establecen naturalmente entre elementos coincidentes semánticamente. Los tipos IV y V tienen una procedencia diferente. Un enfoque complementario del desarrollado, permite visualizar a R_i como una tabla desnormalizada. Las *rirs* que participan de caminos referenciales, se transforman cuando las tablas son ensambladas. Supóngase que R_1 , R_2 y R_3 están vinculadas por un camino de *rirs*, donde $rir_1:R_1[X] \ll R_2 [K_2]$ es de tipo I y $rir_2:R_2 [Y] \ll R_3[K_3]$ es de tipo IV. Cuando se efectúa $(R_1 \text{ JOIN } R_2) \text{ JOIN } R_3$, la rir_1 se convierte en una de tipo IV. Lo mismo ocurre cuando ambas *rirs* son de tipo IV [27, 28]. El caso V presenta esta característica cuando los pares de *rirs* entre R_1 , R_2 y R_3 son los siguientes: (I,V), (IV,V), (III, V), (V, I), (V, IV), (V, III), (V, V).

Esto significa que no hay posibilidad de realizar transformaciones de los esquemas con estas características en forma iterativa con el objeto de alcanzar un punto de normalización,

puesto que no pueden obtenerse restricciones de estos tipos a partir de la desnormalización de tablas bien estructuradas [27, 28].

En algunos casos, la reingeniería del esquema puede conducir hacia uno correcto en alguna de las extensiones del MER, con el soporte de una herramienta y con mínima intervención del usuario. Otros casos son más complicados y, por la dificultad de encontrar un mecanismo, la labor de rediseño se vuelve artesanal.

4 Análisis de las *dis*

Una *dependencia de inclusión (di)* es una expresión $R_i[W] \subseteq R_d[Y]$. Si un atributo o conjunto de ellos (Y) es referenciado vía una *di*, puede suponerse que representa el identificador de una entidad relevante del mundo real. Así, para analizar las *dis*, el énfasis está puesto principalmente en la estructura de la relación R_d .

Dado que el término derecho puede verse como una vista obtenida a partir de dos tablas vía una *rir*, de ahora en adelante se adopta la convención de nombrar el término derecho de la *di*, de acuerdo con el tipo de *rir* que lo genera. Recordando que los tipos de *rirs* son cinco, si la *rir* que da lugar al término derecho es de tipo 't' ($t = I, II, \dots, V$), se mencionará "término derecho de tipo 't' ". Al igual que en secciones anteriores, R_i y R_d son los términos izquierdo y derecho de la restricción. W_i es el conjunto de atributos referenciantes en R_i . Excepto en el caso IV, la estructura de este conjunto de atributos no es relevante. K_d es la clave primaria de R_d . R_d proviene de R_d^1 y R_d^2 que a su vez son las componentes virtuales de R_d ; K_d^1 y K_d^2 son las claves primarias de R_d^1 y R_d^2 respectivamente; K_d^{11} es un subconjunto estricto de K_d^1 ; Z_d^1 es un subconjunto de atributos secundarios en R_d^1 .

Los casos 6. a 10. de la Tabla 1 pueden ser agrupados en dos subcasos: a) los atributos no-clave constituyen una clave alternativa y b) los atributos no-clave no son una clave alternativa. El subcaso a) ha sido desarrollado completamente en [18].

Recuérdese que las expresiones en cursiva resaltadas entre comillas en las que siguen, son un caso particular de *reglas del negocio* denominadas *restricciones de dominio*. En los casos siguientes se verá que el conjunto de valores válidos se materializa a través del término derecho de las *dis*.

4.1 Término derecho de tipo I: $R_i[W_i] \subseteq R_d[K_d]$. (casos 1 a 5 en la Tabla 1; las *rirs* están sintácticamente incluidas en este caso).

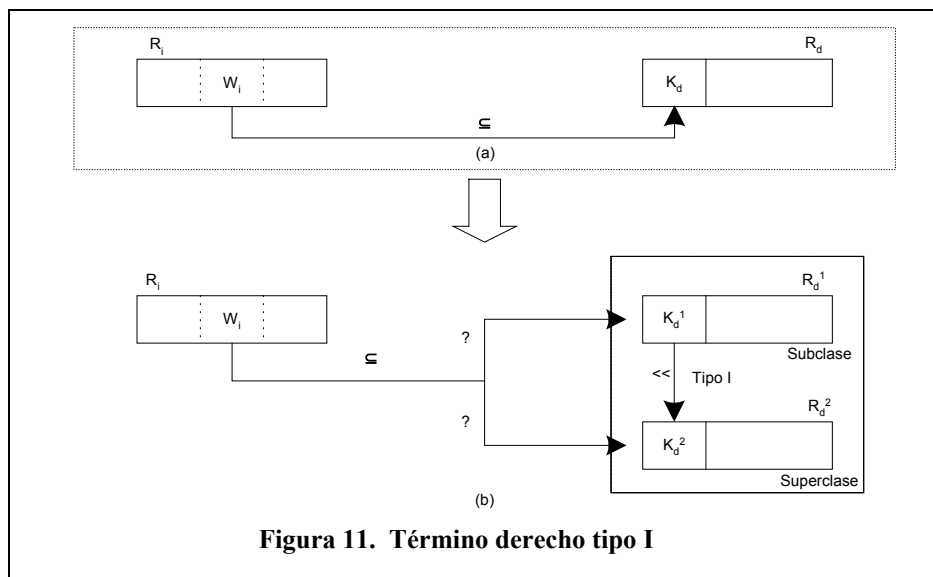
En algunos casos las *dis* pueden lucir como *rirs* pero posiblemente no lo sean. Considérese el siguiente esquema, en el cual la relación SUPERVISOR (R_i) representa los empleados que tienen a su cargo otros empleados a quienes supervisan, y EMPLEADO (R_d) contiene los datos personales y de capacitación de todos los empleados de una organización. De esta forma,

SUPERVISOR (#Empleado, Responsabilidad) y
 EMPLEADO (#Empleado, NombreEmpleado, TítuloTécnico, ÁreadeCapacitación)
 Sea la *di* SUPERVISOR [#Empleado] \subseteq EMPLEADO [#Empleado].

Supóngase que TítuloTécnico y ÁreadeCapacitación son atributos inaplicables, ya que son no nulos sólo cuando un empleado está capacitado como Técnico. Intuitivamente, este caso puede ser rediseñado de la siguiente forma

(R_d^2) EMPLEADO (#Empleado, NombreEmpleado)
 (R_d^1) EMPLEADO-TECNICO (#Empleado, TítuloTécnico, ÁreadeCapacitación)
 y (R_i) SUPERVISOR (#Empleado, Responsabilidad).

Para generalizar el problema considérese el diagrama de la Figura 11. Si R_d es considerada como la unificación de R_d^1 y R_d^2 , la relación entre R_i y R_d a través de $\#Empleado$ parece una *rir*, pero podría estar ocultando una nueva restricción ya que R_d contiene dos entidades diferentes, R_d^1 y R_d^2 , y no se permite realizar referencias específicas a uno de ellas. El diseñador no tiene la alternativa de referenciar a un simple empleado o a un técnico, puesto que la estructura sólo le permite especificar referencias a la tabla unificada.



En consecuencia, la restricción SUPERVISOR [#Empleado] \subseteq EMPLEADO [#Empleado] se visualiza como una *rir*, pero cuando "sólo los empleados técnicos pueden ser supervisores", no es conceptualmente una *rir*. Entonces, si cualquier empleado puede ser supervisor, debe especificarse la siguiente restricción:

EMPLEADO-TECNICO [#Empleado] << EMPLEADO [#Empleado]
y SUPERVISOR [#Empleado] << EMPLEADO [#Empleado].

Por el contrario, si la referencia debe estar basada solamente en los empleados capacitados como Técnicos, las restricciones referenciales correctas son

EMPLEADO-TECNICO [#Empleado] << EMPLEADO [#Empleado]
y SUPERVISOR [#Empleado] << EMPLEADO-TECNICO [#Empleado].

4.2 Término derecho de tipo II: $R_i[W_i] \subseteq R_d[Z_d]$ (casos 6 a 10 en la Tabla 1).

Considérense las siguientes relaciones (claves subrayadas) y la dependencia de inclusión indicada. La relación PRESUPUESTO (R_i) contiene los datos correspondientes a subsidios y PROYECTO (R_d) tiene los datos relativos a proyectos en desarrollo en los departamentos de una organización.

PRESUPUESTO (#Expediente, NombreDepto, Monto) y
PROYECTO (#Proyecto, NombreProyecto, NombreDepto, FechaComienzo)

Sea la *di* PRESUPUESTO [NombreDepto] \subseteq PROYECTO [NombreDepto].

De acuerdo con este diseño, el objeto de la realidad que ha sido omitido es DEPARTAMENTO. Este caso se esquematiza en la Figura 12. Si el diseñador debe especificar que “*sólo aquellos departamentos teniendo proyectos en marcha pueden tener presupuesto asignado*”, la transformación del conjunto de relaciones es

R_d^1 : PROYECTO (#Proyecto, NombreProyecto, NombreDepto, FechaComienzo)
 R_i : PRESUPUESTO (#Expediente, NombreDepto, Monto) y R_d^2 : DEPARTAMENTO (NombreDepto).

El conjunto **DI** resultante es:
 PRESUPUESTO [NombreDepto] << DEPARTAMENTO [NombreDepto]
 y PROYECTO [NombreDepto] << DEPARTAMENTO [NombreDepto].

y la restricción entre comillas es la siguiente *di* remanente (*dir*) expresando la *regla del negocio* mal representada

DEPARTAMENTO [NombreDepto] \subseteq PROYECTO [NombreDepto].

A partir de las últimas dos restricciones, se concluye que PROYECTO.NombreDepto y DEPARTAMENTO.NombreDepto tienen siempre los mismos valores.

Por el contrario, si el diseñador intenta hacer referencia simplemente a cualquier Departamento, la entidad oculta DEPARTAMENTO debiese ser directamente modelada, entonces el esquema rediseñado sería el mismo y la *di* original se reemplazaría con las siguientes *ri*s:

PRESUPUESTO [NombreDepto] << DEPARTAMENTO [NombreDepto]
 y PROYECTO [NombreDepto] << DEPARTAMENTO [NombreDepto].

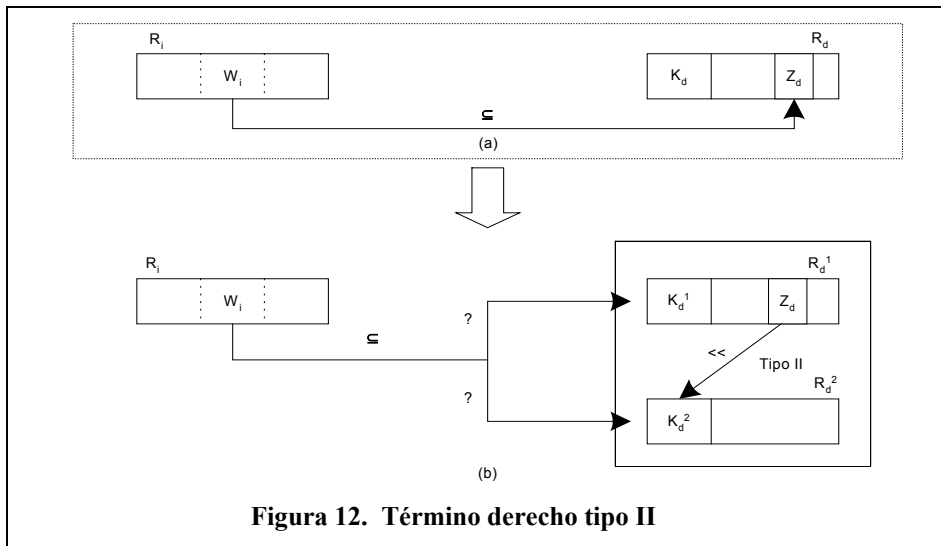


Figura 12. Término derecho tipo II

4.3. Término derecho de tipo III: $R_i[W_i] \subseteq R_d[K_d^1]$ (casos 11 a 15 en la Tabla 1).

Considérense las siguientes relaciones (claves subrayadas) y las dependencias de inclusión listadas a continuación.

CURSO (#Curso, Profesor)
 (R_i) ESTUDIANTE-DELEGADO (Id-estudiante, función)
 (R_d) CURSADA (#Curso, Id-estudiante, Nota)
 CURSADA [#Curso] << CURSO [#Curso]
 ESTUDIANTE-DELEGADO [Id-estudiante] ⊆ CURSADA [Id-estudiante]

Las entidades relevantes para este análisis son ESTUDIANTE-DELEGADO (R_i) y CURSADA (R_d). La primera contiene información relativa a estudiantes que offician como Delegados de sus compañeros y la segunda contiene los datos referidos al desempeño académico de los alumnos en los diferentes cursos. Estas restricciones indicadas sugieren una entidad del mundo real que no ha sido apropiadamente modelada: ESTUDIANTE (R_d²). El diagrama de la Figura 13 muestra este caso. Por cuestiones de notación, K_d¹ en la tabla R_d (a), es renombrada K_d¹² en (b). Naturalmente, K_d¹² ≡ K_d².

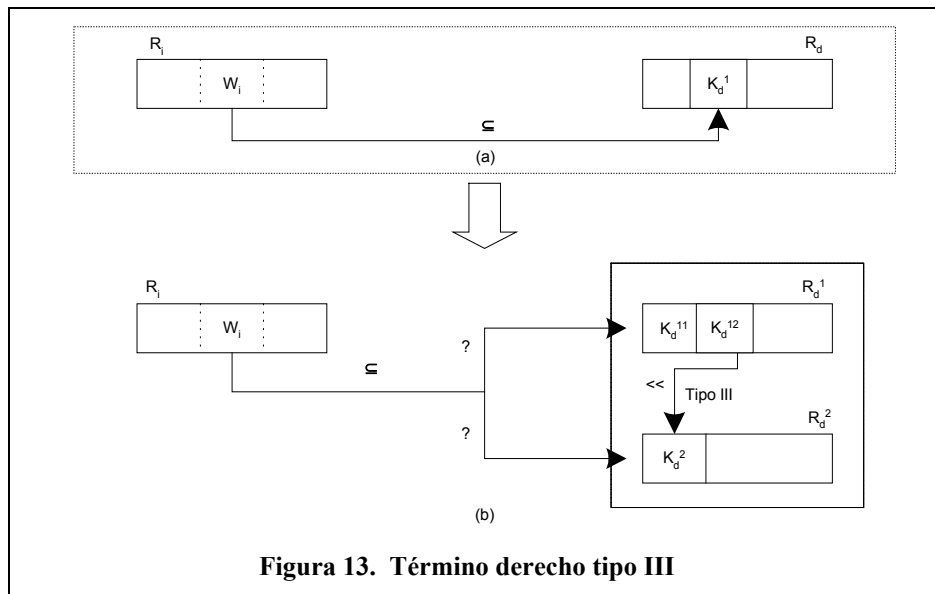
Nuevamente, hay dos escenarios posibles. Si el requerimiento fue sólo establecer el vínculo semántico entre ESTUDIANTE-DELEGADO (R_i), CURSADA (R_d¹) y CURSO, luego las referencias debieron ser:

ESTUDIANTE-DELEGADO [Id-estudiante] << ESTUDIANTE [Id-estudiante]
 CURSADA[Id-estudiante] << ESTUDIANTE [Id-estudiante]
 CURSADA [#Curso] << CURSO [#Curso]

Por otra parte, si el diseñador en realidad deseaba especificar una restricción del estilo “sólo aquellos estudiantes con nota en al menos un curso pueden ser delegados” (es decir, aquéllos que sólo se encuentran representados en CURSADA), la siguiente *di* debería agregarse:

ESTUDIANTE [Id-estudiante] ⊆ CURSADA [Id-estudiante]

Nuevamente la expresión entre comillas es una *regla del negocio* que queda oculta por la imposibilidad de hacer referencia a la entidad apropiada.



4.4 Término derecho de tipo IV: $R_i[W_i] \subseteq R_d[K_d \cup Z_d]$ (casos 16 a 20 en la Tabla 1).

Tampoco en esta situación es posible conocer, sin el conocimiento del diseñador, cuál de las dos relaciones colapsadas en el término derecho es el objetivo de la referencia. Si la decisión del diseñador fue referenciar R_d^2 , los conjuntos correctos de relaciones y restricciones deben ser reformulados en forma análoga a los casos anteriores. Este caso tiene una característica particular. Considérese el siguiente ejemplo, adaptado de [22] (claves subrayadas):

DIRECCIÓN (IdDirector, IdDepto) y NÓMINA (IdProfesor, IdDepto)

Y la dependencia de inclusión:

DIRECCIÓN [IdDirector, IdDepto] \subseteq NÓMINA [IdProfesor, IdDepto]

Indicando que el director de un departamento debe ser un integrante de la nómina de profesores de ese mismo departamento. Nótese que a partir de la dependencia de inclusión y de la dependencia funcional NÓMINA: $\text{IdProfesor} \rightarrow \text{IdDepto}$ puede deducirse la segunda dependencia funcional de este esquema DIRECCIÓN: $\text{IdDirector} \rightarrow \text{IdDepto}$.

Este ejemplo sencillo ilustra un error de diseño bastante frecuente. Las consecuencias de este mal diseño son dos: la interacción de dependencias funcionales y de inclusión puede dar lugar a restricciones de integridad derivadas que el diseñador no tenía previstas; por otra parte, algún atributo puede ser redundante tal como ocurre en este caso con el IdDepto en DIRECCIÓN, que puede ser inferido directamente pues en su rol de profesor, el director ya tiene un departamento asignado.

Formalmente este caso se resuelve por aplicación de la regla de inferencia “pullback” [6] reproducida en [22] que indica: Dadas dos relaciones $R_i(X,Y)$ y $R_d(Z,W)$ en las que se verifica la dependencia de inclusión $R_i[XY] \subseteq R_d[ZW]$ y la dependencia funcional $R_d: Z \rightarrow W$, siendo $|X|=|Z|$ (ambas cardinalidades iguales), entonces se puede inferir $R_i: X \rightarrow Y$. La dependencia funcional inferida ayuda a esclarecer la estructura del término izquierdo de la *di* revelando en algunos casos una entidad oculta, es decir, cierto grado de desnormalización. La dependencia de inclusión original debió plantearse:

DIRECCIÓN [IdDirector] \subseteq NÓMINA [IdProfesor]

4.5 Término derecho de tipo V: $R_i[W_i] \subseteq R_d[K_d^1 \cup Z_d]$ (casos 21 a 25 en la Tabla 1).

Este caso es similar a los casos II y III y también lo son los escenarios alternativos. Por esta razón el esquema correcto y el conjunto de restricciones pueden ser obtenidos en forma análoga.

4.6 Comentarios finales sobre *dis*

Es importante notar que el diseño normalizado podría ser deducido a través del posible origen de R_d , es decir, sobre la base de la sintaxis de la unificación. Sin embargo, esto es insuficiente porque la intención real del usuario debería ser adivinada. Esto significa que la semántica debe proveerla quien conoce el UdeD. Las presuntas *reglas del negocio* ocultas, deben confirmarse efectuando un rastreo retrospectivo en el UdeD, habiendo sido elicidadas de la manera apropiada. En otras palabras, los aspectos sintácticos guían el proceso de conversión y entendimiento de las estructuras, pero previamente, el diseñador debe elucidar los aspectos

semánticos. Debe suponerse que el término derecho es un ensamble de dos entidades (regulares o débiles).

La estructura del término izquierdo define el tipo de *rir* que resulta de la aplicación de la heurística (Sección 5.) en los casos I, II, III y V. El tipo IV siempre debe resolverse con la regla de “pullback”. Esto promueve la modificación de la estructura de R_i y eleva su grado de normalización. Como una consecuencia adicional, las *dis* originales se especifican como *rirs* cuyos tipos devienen del tipo de término izquierdo en los casos I, II y III. En los casos IV y V resultan *rirs* de tipos I y III respectivamente.

Si R_i es de tipo V y la *di* es de tipo IV puede obtenerse otro esquema final si la correspondencia de atributos fuese la inversa. Esta situación corresponde al caso atípico de la Forma Normal de Boyce - Codd [10, 30] y la conversión resulta en dos *rirs* de tipo I. Finalmente, en el análisis del tipo V, debe suponerse que $K_d^2 \rightarrow Z_i$.

5 Conversión de Esquemas con *dis*

En las secciones previas se han considerado los diferentes casos de *dis* y los casos especiales de *rirs* incluyendo el análisis de sus posibles orígenes y en el significado del vínculo entre las tablas involucradas. Esta sección presenta una heurística para convertir el esquema real con *dis* en uno equivalente con *rirs* y *dirs*. En lo que sigue, considérense las referencias detalladas en la Tabla 2.

Sean $\mathbf{DI} = \{ \dots; R_{ij}[W_{ij}] \subseteq R_d[W_d]; \dots \}$ $j=1, 2, \dots$ y $\mathbf{DF} = \{ \dots, R_i:K_i \rightarrow \text{atr}(R_i)-K_i; R_d:K_d \rightarrow \text{atr}(R_d)-K_d; R_d: W_d \rightarrow Z_d^2; \dots \}^{\dagger\dagger}$. En los casos II a V, la última dependencia funcional es secundaria. En el caso I esta dependencia revela que Z_d^2 es un conjunto de atributos inaplicables. El caso IV no se incluye pues debe ser resuelto con la regla de “pullback”. Sea el esquema normalizado $\mathbf{R}^n = \langle \mathbf{R}^n, \mathbf{D}^n \rangle$, con $\mathbf{D}^n = \{ \mathbf{DF}^n, \mathbf{DI}^n \}$. Se asume que Z_d^2 o un subconjunto estricto de él no forman parte de otra clave extranjera. En caso contrario, es necesario considerar lo siguiente: a) si la clave extranjera (o referencia pura) está formado por atributos de K_{NUEVA} y/o Z_d^2 , directamente relocalizarla en NUEVA; b) si está formada por atributos de Z_d^2 y de R_d^1 , es necesario un nuevo análisis de su estructura, en el marco de su rol como término izquierdo de una *rir* atípica.

TIPO	atr(R_d)		W_d
	ATRIBUTOS NO-CLAVE	K_d	
I	$Z_d^1 \cup Z_d^2$	K_d	K_d
II	$Z_d \cup Z_d^1 \cup Z_d^2$	K_d	Z_d
III	$Z_d^1 \cup Z_d^2$	$K_d^1 \cup K_d^2$	K_d^1
V	$Z_d \cup Z_d^1 \cup Z_d^2$	$K_d^1 \cup K_d^2$	$K_d^1 \cup Z_d$

Tabla 2: componentes de $\text{atr}(R_d)$. W_d es el atributo referenciado (simple o compuesto). K_d^1 y K_d^2 son subconjuntos estrictos y disjuntos de K_d . Z_d , Z_d^1 y Z_d^2 son subconjuntos disjuntos de $\text{atr}(R_d) - K_d$. Z_d^1 y Z_d^2 pueden ser vacíos.

^{††} Sin pérdida de generalidad, se consideran esquemas con sólo una relación derecha desnormalizada R_d .

Los siguientes pasos conducen el proceso de reingeniería:

Input: Un esquema relacional $\mathbf{R} = \langle \mathbf{R}, \mathbf{D} \rangle$ con entidades y relaciones ocultas;

Output: Un esquema relacional $\mathbf{R}^n = \langle \mathbf{R}^n, \mathbf{D}^n \rangle$ normalizado;

For Cada entidad oculta detectada según el análisis del término derecho $R_d[W_d]$ de *dis* en **DI**,

crear e incorporar a \mathbf{R}^n una relación NUEVA con:

$K_{NUEVA} \equiv W_d$;

If Existen *dfs* con la estructura $R_d: W_d \rightarrow Z_d^2$ /* Hay atributos descriptivos que corresponden a la entidad oculta en R_d */ ----- (1)

Then Transferirlos como atributos descriptivos en NUEVA;

/* atr(NUEVA) = W_d, Z_d^2 */

Eliminarlos de R_d , convirtiendo R_d en R_d'

/* atr(R_d') = K_d, Z_d^1 */

EndIf:

Case II, III or V:

For Cada *di* definida $R_{i,j}[W_{i,j}] \subseteq R_d[W_d]$ $j=1, 2, \dots$ /* el término izquierdo es el conjunto de atributos que identifican la entidad omitida en NUEVA */

Redefinir la *di* como $R_{i,j}[W_{i,j}] \ll NUEVA[K_{NUEVA}]$ ----- (2)

EndFor:

Incluir $R_d'[W_d] \ll NUEVA[K_{NUEVA}]$ en **DIⁿ**; ----- (3)

If existía una *regla del negocio oculta*

Then Agregar una *dir* $NUEVA[K_{NUEVA}] \subseteq R_d'[W_d]$ a **DIⁿ** ----- (4)

EndIf:

Case I:

For Cada *di* definida $R_{i,j}[W_{i,j}] \subseteq R_d[W_d]$ $j=1, 2, \dots$ /* el término izquierdo es el conjunto de atributos que identifican la entidad omitida en NUEVA */

If Representa una *regla del negocio oculta*

Then Redefinir la *di* como $R_{i,j}[W_{i,j}] \ll NUEVA[K_{NUEVA}]$

Else Incluir $R_{i,j}[W_{i,j}] \ll R_d'[K_d]$ en **DIⁿ**

EndIf: ----- (2)

If NUEVA y R_d' están vinculadas por una relación *1:1 mandatoria*

Then Incluir $R_d'[W_d] \ll NUEVA[K_{NUEVA}]$ en **DIⁿ** ----- (3)

EndIf:

Agregar una *dir* $NUEVA[K_{NUEVA}] \subseteq R_d'[W_d]$ a **DIⁿ** ----- (4)

EndFor:

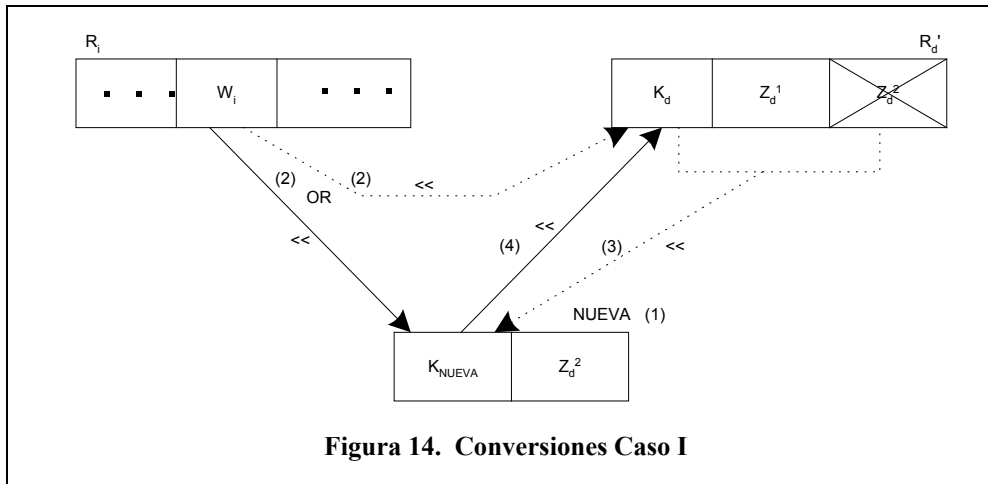
End.

La aplicación de estos pasos resulta en: $\mathbf{R}^n = \mathbf{R} \cup \{NUEVA; R_d'\} - \{R_d\}$.

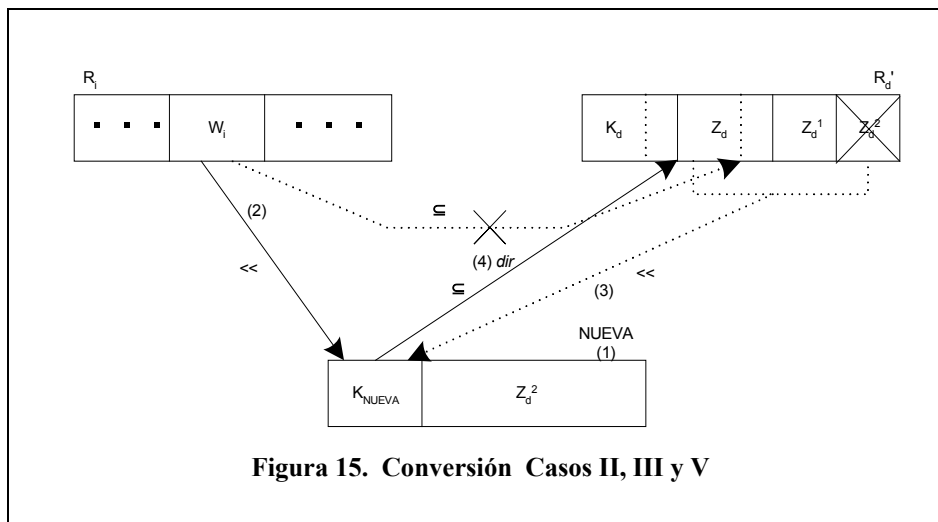
Para los casos II, III y V, $\mathbf{DI}^n = \mathbf{DI} \cup \{R_d'[W_d] \ll NUEVA[K_{NUEVA}]\} \cup \{R_{i,j}[W_{i,j}] \ll NUEVA[K_{NUEVA}]\}_{j=1,2,\dots} - \{R_{i,j}[W_{i,j}] \subseteq R_d[W_d]\}_{j=1,2,\dots}$. Para el caso indicado en (4) $\mathbf{DI}^n = \mathbf{DI} \cup \{R_d'[W_d] \ll NUEVA[K_{NUEVA}]; NUEVA[K_{NUEVA}] \subseteq R_d'[W_d]\} \cup \{R_{i,j}[W_{i,j}] \ll NUEVA[K_{NUEVA}]\}_{j=1,2,\dots} - \{R_{i,j}[W_{i,j}] \subseteq R_d[W_d]\}_{j=1,2,\dots}$.

$DF^n = DF \cup \{NUEVA:K_{NUEVA} \rightarrow Z_d^2\} - \{R_d:K_d \rightarrow Z_d^2, R_d:W_d \rightarrow Z_d^2\}$. Para el caso I, DI^n y DF^n pueden ser obtenidos de manera similar.

Las siguientes figuras esquematizan los procesos de conversión para los casos I y II, III y V respectivamente. Por razones de claridad en la ilustración, las relaciones se han representado mediante una adaptación de la notación gráfica de la Fig 1.



En resumen, un mecanismo de conversión está destinado a obtener una correspondencia 1-1 entre tipos de objetos del UdeD y esquemas de relaciones. Esto facilita la conexión y rastreo de las nociones involucradas en ambos mundos, el real (UdeD) y el de la base de datos. De esta forma el proceso de conversión está guiado por las siguientes preguntas: a) qué concepto del UdeD representa W_d ?; b) cuál es la representación apropiada de W_d ?; c) cuál es la manera correcta de representar las relaciones con W_d ?; d) hay algún atributo o conjunto de ellos en R_d que sólo dependa funcionalmente de W_d ?; e) cuál es el significado de la *di* relacionada con W_d ? El método presentado ayuda a encontrar las respuestas en el UdeD mientras guía la conversión del esquema en otro semánticamente enriquecido.



5.1 Ejemplos de las secciones 4.1 y 4.2 revisitados

A continuación se reconsideran formalmente los dos primeros ejemplos de la Sección 4.

Término derecho tipo I (Ejemplo de la Sección 4.1)

Sea un esquema relación con la relaciones SUPERVISOR (#Empleado, Responsabilidad) (R_i) y EMPLEADO (#Empleado, NombreEmpleado, TítuloTécnico, ÁreaCapacitación) (R_d) y la *di* SUPERVISOR [#Empleado] \subseteq EMPLEADO [#Empleado].

Luego, $W_d \equiv K_d = \#Empleado$; $Z_d^2 = \text{TítuloTécnico, ÁreaCapacitación}$ y $Z_d^1 = \text{NombreEmpleado}$.

Siguiendo los pasos de la heurística:

- (1) EMPLEADO-TECNICO es la entidad NUEVA, entonces $R^n = \{\text{EMPLEADO}';$
 $\text{EMPLEADO-TECNICO}; \text{SUPERVISOR}\}$. $R_d' \equiv \text{EMPLEADO}'$; $\text{atr}(\text{EMPLEADO}') =$
 $\#Empleado, \text{NombreEmpleado}$ y $\text{atr}(\text{NUEVA}) = \#Empleado, \text{TítuloTécnico},$
 ÁreaCapacitación .
- (2) Si la referencia debe estar basada solamente en los empleados capacitados como Técnicos, agregar la referencia SUPERVISOR [#Empleado] \ll EMPLEADO-TECNICO [#Empleado] a **DI**ⁿ.
Si cualquier empleado puede ser supervisor, debe especificarse:
SUPERVISOR[#Empleado] \ll EMPLEADO' [#Empleado].
- (3) No es aplicable
- (4) Agregar la referencia EMPLEADO-TECNICO [#Empleado] \ll EMPLEADO' [#Empleado] a **DI**ⁿ.

Término derecho tipo II (Ejemplo de la Sección 4.2)

Séa el esquema relacional PROYECTO (#Proyecto, NombreProyecto, NombreDepto, FechaComienzo) (R_d) y PRESUPUESTO (#Expediente, NombreDepto, Monto) (R_i); y la *di* PRESUPUESTO [NombreDepto] \subseteq PROYECTO [NombreDepto]. Luego, $W_d \equiv Z_d = \text{NombreDepto}$; $Z_d^2 = \emptyset$ y $Z_d^1 = \text{NombreProyecto, FechaComienzo}$.

- (1) DEPARTAMENTO es la entidad del mundo real que ha sido omitida (NUEVA), entonces $R^n = \{\text{PRESUPUESTO}, \text{PROYECTO}', \text{DEPARTAMENTO}\}$. En este caso $\text{atr}(\text{DEPARTAMENTO}) = \text{NombreDepto}$. PROYECTO' se especifica con los mismos atributos que PROYECTO.
- (2) La *di* debe reformularse como la siguiente *rir*: PRESUPUESTO [NombreDepto] \ll DEPARTAMENTO [NombreDepto] agregándola a **DI**ⁿ.
- (3) La siguiente *rir* de incluirse en **DI**ⁿ: PROYECTO' [NombreDepto] \ll DEPARTAMENTO [NombreDepto].
- (4) Si se debe registrar que "solo aquéllos departamentos con al menos un proyecto pueden tener presupuesto asignado" la siguiente *dir* debe incluirse en **DI**ⁿ: DEPARTAMENTO [NombreDepto] \subseteq PROYECTO'[NombreDepto] (flecha punteada en la Figura 15).
Si el requisito es solo establecer el vínculo semántico entre PRESUPUESTO, DEPARTAMENTO, y PROYECTO', el proceso de reingeniería está terminado.

5.2 Materialización de las *dirs*

Como se ha puntualizado, las *reglas del negocio* ocultas, reveladas en las secciones anteriores constituyen *restricciones de dominio*, controlando los valores de un atributo respecto de los

elementos de un conjunto de valores válidos, definido en estos casos como los valores de las columnas de $R_d[Y]$. Las operaciones de actualización podrían modificar este conjunto. Una implementación conceptual debería calcular los valores permitidos antes de usarlos, a pesar de que razones de tiempo naturalmente sugieren que una solución mejor es calcularlo incrementalmente. Este enfoque es práctico y sencillo si el diseñador no le da a este conjunto el status de tabla en el esquema. Si este conjunto es preservado en el esquema en la misma forma que cualquier otra tabla, deben implementarse otras formas de mantenimiento.

Aunque existen diferencias conceptuales, cláusulas CHECK generales o código especialmente escrito (triggers, stored procedures o código en las aplicaciones) pueden resolver el problema de mantener el conjunto en estado consistente.

La primera solución es bastante directa y natural. Para el ejemplo de la sección anterior, la restricción puede especificarse:

```
CHECK (DEPARTAMENTO.NombreDepto IN (SELECT NombreDepto FROM
PROYECTO))
```

En general, para los casos II, III y V:

```
CHECK (NUEVA.KNUEVA IN (SELECT Rd.Wd FROM Rd)).
```

Esta opción es declarativa, y es aplicable siempre que la acción deseada como respuesta a toda actualización que la violente sea el rechazo (rollback). En el caso I, esta materialización es directa porque todas las restricciones pueden ser especificadas como *rirs*, aún en la situación explicada en el paso (3), aunque se requeriría la especificación del ciclo referencial entre NUEVA y R'_d , con las dificultades inherentes a la entrada en régimen de ambas tablas.

Las soluciones programables permiten implementar otras acciones referenciales diferentes al rechazo, en relación con la tabla NUEVA. Nótese que en este escenario, NUEVA no puede ser modificada por operaciones directas, cuando hay involucrada una *regla del negocio* oculta. A continuación se mencionan sucintamente algunos aspectos relativos a las acciones referenciales reparadoras.

Inserciones o actualizaciones en R_d pueden promover una inserción en NUEVA (inserciones en Cascada) *si y sólo si el valor del atributo en cuestión (W_d) es nuevo*. En otro caso la operación debe ser rechazada. Las inserciones están prohibidas en cualquier otro caso. Estas acciones referenciales deben ser modeladas de acuerdo a las respectivas especificadas para la *di* original.

Por otra parte, cuando el borrado o actualización en R_d *afecta la última instancia del atributo referido (W_d)*, debe provocar el borrado de la tupla que contiene el mismo valor en NUEVA, o bien la operación debe ser rechazada. Una vez más, estas acciones referenciales deben ser especificadas de acuerdo a las correspondientes planeadas para la *di* original.

Las operación de modificación (en forma análoga a lo que ocurre con las supresiones) tiene una semántica especial ya que debe ser entendida en una de dos formas: i) representa la actualización de una tupla particular de R_d o ii) representa la actualización de la entidad empotrada (ahora en NUEVA). En ambos casos, la inclusión mutua debe ser mantenida, siempre que sea necesaria la *dir* (casos II, III, y V) y cuando existe la relación 1:1 entre NUEVA y R'_d .

Una inserción en R_i *debería ser restringida si la instancia de W_i no está en K_{NUEVA}* . Si la acción referencial de la *di* original tenía la semántica de las acciones 'set default' o 'set null', esta semántica debe ser preservada en la nueva *rir*. 'Cascada' no es una acción válida dado que genera nulos en K_{NUEVA} en todos los casos salvo el IV. Las supresiones en R_i no traen problemas de integridad y por último, si están permitidas, las operaciones sobre NUEVA propagables a R_i , deberían reaccionar copiando las acciones referenciales especificadas originalmente para la *di*.

En el caso de las dependencias de inclusión puras, las acciones referenciales para las operaciones sobre la tabla derecha deben definirse en dos escenarios diferentes. Puesto que la

existencia de una *di* revela la presencia de una entidad empotrada en otra, debe entenderse sobre cual de las dos se efectúa la operación de actualización:

- i) si se trata de entidades continentales, una baja “arrastra” a la entidad empotrada, y produce la propagación a la relación izquierda en caso de estar suprimiendo la última instancia de la entidad empotrada. Una modificación, salvo en el caso I, no tiene efectos sobre las entidades empotradas;
- ii) por el contrario una baja de una entidad empotrada resulta en una modificación masiva de todas las instancias correspondientes en las entidades continentales que se manifiesta como un anulado de los atributos, mientras que una modificación se visualiza prácticamente como el cambio de una instancia de identificador de la entidad empotrada, que obviamente puede tener repeticiones.

6 Relación con otros Trabajos

Textos sobre bases de datos tales como [1, 9, 10, 15, 29] proveen un amplio tratamiento de las dependencias de inclusión y referencias, tanto en los aspectos prácticos como en los teóricos. El artículo en [11] puede considerarse como un análisis fundacional de la integridad referencial.

Muchos proyectos de investigación han sido dedicados al desarrollo de metodologías y heurísticas con el objeto de encarar el mejoramiento del nivel semántico del esquema conceptual [3, 4, 5, 7, 12, 16, 17, 18, 19, 20, 22, 23, 24, 25, 26, 27, 28, 29].

En [5] se describe una estrategia de optimización en dos fases para esquemas relacionales conteniendo dependencias de inclusión basadas en claves, teniendo en cuenta la preservación de tales dependencias en función de sus acciones referenciales para inserciones y borrados.

La metodología presentada en [23] y [24] asume que los esquemas están normalizados, es decir al menos en 3FN o FNBC. En [24] se propone un método formal para capturar la semántica estructural de los sistemas de información. Puede ser usada tanto para analizar la semántica de sistemas relacionales existentes como para convertir esquemas relacionales en esquemas orientados a objetos. El método considera esquemas de relación, dependencias funcionales y restricciones de integridad referencial. Permite la derivación de esquemas Entidad-Relación en esquemas relacionales convencionales, pero está restringida a una clase particular de los mismos.

Castellanos [7] presenta una metodología para el enriquecimiento semántico consistente en dos fases, la primera correspondiente a la adquisición de conocimiento y la segunda al proceso de conversión del esquema. En este artículo la autora considera las dependencias de inclusión desde una perspectiva más general, analizando un conjunto de casos basados en la composición de sus términos izquierdo y derecho, siendo este análisis la base para un proceso de reingeniería semiautomático que reconoce estructuras ocultas. Johannesson [18] desarrolló un método que traduce un esquema en un modelo de datos tradicional en un esquema conceptual normalizado, con la misma capacidad de información que el esquema original. La propuesta está basada en una extensión del MER que incluye el concepto de subtipo. Su clasificación del término derecho es incompleta ya que conformaciones tales como “parte-de-clave”, “clave+atributos secundarios” y “parte-de-clave+atributos-secundarios” no han sido considerados. Los trabajos de Johannesson [18] y Castellanos [7] tratan los mismos casos de dependencias de inclusión puras y restricciones de integridad referencial.

En [27, 28] se analizan las diferentes configuraciones de las restricciones de integridad referencial, y las consecuencias de la actualización de relaciones desnormalizadas, originalmente vinculadas por los diferentes tipos de *rirs*. Estos trabajos extienden los resultados de [23] ya que se caracterizan e incluyen en el análisis dos casos patológicos adicionales, los tipos IV y V. En [25, 26] se presentan resultados preliminares del análisis

desarrollado en este trabajo, incluyendo sólo el análisis de los diferentes casos de *dis*, y una versión simplificada de la heurística de conversión.

En [22] se presenta un análisis exhaustivo de las dificultades que la redundancia causa en el mantenimiento de la consistencia cuando las dependencias de inclusión puras interfieren con las dependencias funcionales, enfatizando la necesidad de eliminar la redundancia y las anomalías de actualización y la preservación de la integridad. En este contexto, se ponen de manifiesto malas prácticas que conducen a diseños incorrectos, incluyendo en los ejemplos de motivación la aplicación de la regla de pullback.

Dey y otros [12] describen un marco general de análisis de las relaciones de orden mayor que 2, con el fin de detectar representaciones inapropiadas. El análisis incluye las cardinalidades máxima y mínima, el grado de la relación, la naturaleza recursiva de las relaciones y las restricciones entre relaciones. Los autores proponen reglas generales para representar relaciones n-arias en el modelo relacional, facilitando por este medio la extracción de más información del dominio de aplicación, lo que resulta en su correcta representación en el diseño conceptual.

Por su parte, en [19, 20] se estudian las características de las combinaciones de relaciones ternarias y binarias semánticamente vinculadas con el objetivo de proveer reglas que permitan detectar, mediante el análisis de las dependencias funcionales, las potenciales descomposiciones de relaciones ternarias en múltiples binarias sin redundancia y semánticamente equivalentes. Este estudio es completado en [3, 4]. En este par de artículos se examinan los posibles patrones de transformación para obtener conjuntos de relaciones en 3FN equivalentes a las relaciones múltiples originales, mediante el agregado de claves candidatas, dependencias de inclusión y otras restricciones de inclusión específicas.

En [16], Henrard y Hainaut presentan el análisis y comparación de varias técnicas para la elicitación de estructuras y dependencias ocultas, destacando la importancia del descubrimiento de las restricciones y poniendo de manifiesto las técnicas y herramientas que ayudan en este proceso. En [17] los autores se concentran en la reconstrucción del esquema DBMS-dependiente (lógico), enfocando en los razonamientos y procesos a través de los cuales estructuras ocultas y restricciones pueden ser elicidadas. Presentan una metodología genérica aplicable al proceso de extracción de estructuras de datos.

En [29] se trata el problema de recuperar esquemas orientados a objetos, a partir de bases de datos relacionales, describiendo una metodología para la reingeniería de los esquemas que está basada en el concepto que los autores denominan 'correlación de claves y datos'. La correlación de claves se corresponde con una visión similar a la que en este trabajo se ha esquematizado en la Figura 2 y representa algunas de las diferentes posiciones de una clave extranjera con respecto a la clave primaria de su relación. La correlación de datos se refiere a las inclusiones o solapamientos parciales de instancias de las relaciones.

7 Conclusión y Trabajo Futuro

Este trabajo extiende resultados conocidos analizando las restricciones de inclusión cuyos componentes están desnormalizados. Los posibles orígenes de las entidades empotradas son esclarecidos y se propone una heurística para la completa reingeniería del esquema. El restablecimiento de las entidades ocultas detectadas permite su remoción de aquellas relaciones que las contienen. Esta acción mejora el nivel de normalización del sistema transfiriendo los atributos descriptivos a la tabla NUEVA y promueve la generación de nuevas restricciones de integridad referencial. En algunas situaciones es posible detectar reglas del negocio enmascaradas como dependencias de inclusión puras irreducibles. En estos casos, la heurística sólo clarifica el problema pero le deja al usuario la resolución del problema en la manera apropiada.

La inferencia de las acciones referenciales de las *dis* iniciales en las correspondientes a las *rirs* y *dirs* producto de la transformación, ha sido sólo bosquejada, requiriendo aún un análisis más detallado. Cuando una *di* irreducible debe permanecer en el esquema, se establecen las condiciones contextuales para su mantenimiento, promoviendo la definición de una regla del negocio bajo la forma de una aserción específica.

En este punto surge inmediatamente una pregunta: si el problema está relacionado con esquemas desnormalizados, ¿porqué no normalizar y concluir de esta forma con el problema? En diferentes contextos, esta pregunta puede tener diferentes respuestas, dependiendo del grado de comprensión del esquema. En bases de datos totalmente documentadas y bien construidas, la mejor solución es normalizar hasta alcanzar el nivel aconsejado por los requerimientos de tiempo de respuesta y espacio de almacenamiento. En un esquema pobremente diseñado, es precisamente la heurística descrita en este artículo la que facilita el descubrimiento y correcto diseño del conocimiento implícito. Nótese que se requiere la determinación completa de los aspectos semánticos con el objeto de alcanzar decisiones relacionadas con el proceso de reingeniería. No obstante esto, el proceso de conversión está guiado por cuestiones sintácticas. Cuanto más amplio sea el marco de referencia de este análisis, más rica la variedad de aspectos semánticos que podrán ser extraídos del modelo conceptual y ser especificados de la forma apropiada en el esquema relacional. La identificación plena de estas nociones permitirá su rastreo desde el UdeD, enriqueciendo todos los estados intermedios en el ciclo de diseño del producto de software.

Con base en las transformaciones presentadas, se proyecta el diseño e implementación de una herramienta computacional interactiva, basada en los casos de conversión presentados, que facilite al usuario la obtención de un esquema de mejor calidad.

Agradecimientos. A los revisores anónimos que con sus valiosos comentarios han contribuido al mejoramiento de este trabajo. Este proyecto es parcialmente financiado por un subsidio de la SeCAT – UNCPBA.

Referencias

- [1] Abiteboul S, Hull R, Vianu V. Foundations on Databases. Addison Wesley Publ. Co. 1995.
- [2] Biskup J. Achievements of Relational Database Schema Design Theory Revisited, Lecture Notes in Computer Science 1358, 29-54. Springer, 1998.
- [3] Camps R. From Ternary Relationship to Relational Tables: A Case Against Common Beliefs, ACM/SIGMOD-Record, 31(2) 46-49. 2002.
- [4] Camps R. Transforming N-Ary Relationships to Database Schemas: An Old and Forgotten Problem. Research Report LSI-02-5-R. Universidad Politécnic de Catalunya, España, 2002.
- [5] Casanova M, Tucherman L, Furtado A, Braga A. Optimization of Relational Schemes Containing Inclusion Dependencies. Proceedings 15th VLDB Conference 317-325. Amsterdam. 1989.
- [6] Casanova M, Fagin R, Papadimitriou C. Inclusion Dependencies and their Interaction with Functional Dependencies, J. Computer and System Sciences, 28, 29-59. 1984.
- [7] Castellanos MG. A Methodology for Semantically Enriching Interoperable Databases. Proceedings 11th British National Conference on Databases 58-75. Keele, 1993.
- [8] Chen P P. The Entity-relationship Model: Toward a Unified View of Data. ACM Trans. on Database Systems. 1(1) 9-36. 1976.
- [9] Codd E. The Relational Model for Database Management. Version 2. Addison Wesley Publ.Co, Reading, MA, 1990.

- [10] Date C. *An Introduction to Database Systems*. Addison Wesley, Longman Pub. Co., Boston, MA, 2000.
- [11] Date C. Referential Integrity. In *Relational Databases, Selected Writings*. Addison Wesley, Reading, MA, 1986, pp. 41-63.
- [12] Dey D, Storey C, Barron T. Improving Database Design through the Analysis of Relationships, *ACM TODS* 24(4) 453-486. 1999.
- [13] Dahchour M, Pirotte A. The Semantics of Reifying N-ary Relationships as Classes. *Proceedings 4th Int. Conf. on Enterprise Information Systems, ICEIS'02*, 580-586, Universidad de Castilla-La Mancha Ciudad Real, Spain, 2002.
- [14] Doorn J, Rivero L. Normalization of Non-BCNF Relations Integrity Constraints. *Proceedings XII International Conference of Systems Engineering, ICSE-97*, 217-222. Coventry, UK, 1997.
- [15] Elmasri R, Navathe S. *Fundamentals of Database Systems*. Benjamin Cummings, Redwood City, 1994.
- [16] Henrard J, Hainaut J-L. Data Dependency Elicitation in Database Reverse Engineering. *Published CSMR 2001* 11-19. Lisbon, Portugal, March 2001.
- [17] Henrard J, Hainaut J-L, Hick J-M, Roland D, Englebort V. Data Structure Extraction in Database Reverse Engineering. *Proceedings of REIS'99 Workshop*. Paris, November 1999. Research paper RP-99-007. Institut d'Informatique. University of Namur.
- [18] Johannesson P. A Method for Transforming Relational Schemas into Conceptual Schemas. *IEEE Trans. on Software Eng* 190-201, 1994.
- [19] Jones T, Song I-Y. Analysis of Binary/Ternary Cardinality Combinations in Entity-Relationship modeling. *Data & Knowledge Engineering*, 19(1) 39-64, 1996.
- [20] Jones, T., Song, I-Y. Binary Equivalents of Ternary Relationships in Entity-Relationship Modeling: a Logical Decomposition Approach. *Journal of Database Management*, 11(2) 12-19, 2000.
- [21] Kolp M, Zimányi E. Relational Database Design using an ER Approach and Prolog. *Proceedings 6th International Conference on Information Systems and Management of Data (CISMOD'95)* 214-231. Bombay, India, November 1995.
- [22] Levene M, Vincent M. Justification for Inclusion Dependency Normal Form. *IEEE TKDE* 12(2) 281-291. March/April, 2000.
- [23] Markowitz V, Makowsky J. Identifying Extended Entity-relationship Object Structures in Relational Schemas. *IEEE Trans. on Software Eng*. 16(8) 777-790, 1990.
- [24] Markowitz V. Referential Integrity Revisited: an Object-oriented Perspective. *Proceedings of the 16th VLDB Conference*, 578-589. Brisbane, Australia, Morgan Kaufman, San Mmateo, CA, 1990.
- [25] Rivero L, Doorn J, Ferraggine V. Elicitation and Conversion of Hidden Objects and Restrictions in a Database Schema. *Proceedings 2002 ACM Symposium on Applied Computing*, 463-469. Madrid, Spain, 2002.
- [26] Rivero L, Doorn J, Ferraggine V. Inclusion Dependencies. In *Developing Quality Complex Database Systems: Practices, Techniques and Technologies* 261-278. Idea-Group Publishing, Hershey, PA. Editor Dr. Shirley A. Becker. Florida Institute of Technology, 2001.
- [27] Rivero L, Doorn J. Integridad Referencial y Actualizaciones en Relaciones Desnormalizadas. *Proceedings XIV Conferencia Latinoamericana de Informática - CLEI Panel 98*, 911-921. Quito, Ecuador, 1998.
- [28] Rivero L, Doorn J. Managing Referential Integrity and Non Key-based Dependencies in a Denormalized Context, *Proceedings 2000 IRMA International Conference* 883-886. Anchorage, Alaska, 2000.

- [29] Tari Z, Buhkres O, Stokes J, Hammoudi S. The Reengineering of Relational Databases based on Key and Data Correlations. In Searching for Semantics: Datamining, Reverse Engineering, etc. Proceedings 7th IFIP 2.6 Working Conference on Database Semantics (DS-7) 183-214. S. Scappapietra y F. Maryanski editores. Chapman & Hall, Lausanne, 1998.
- [30] Ullman J. Principles of Database and Knowledge-Base Systems. Vol. 1. Computer Science Press, NY, 1988.
- [31] Wand Y, Storey V, Weber R. An Ontological Analysis of the Relationship Construct in Conceptual Modeling, ACM TODS, 24(4) 494-528. 1999.