

**Modelado de Datos en el Mundo Real:
Teoría vs. Práctica**

**Tema 6:
Dependencias de Comparación de Conjuntos de Valores
y otras Transformaciones**

Laura Rivero, Viviana Ferraggine
2010 Grupo Base de Datos y Procesamiento de Señales

Dependencias de Comparación de Conjuntos de Valores (DCCV).

- Caracterización de las Dependencias:
 - De inclusión (*DI*)
 - De exclusión (*DEX*)
 - De igualdad (*DIG*)
 - De superposición parcial (*DSP*)
- Formalización.
- Posibilidad de su representación directa o mediante conversión en un modelo de datos.
- Patrones declarativos de especificación en casos de imposibilidad de modelado.

DCCV y Reglas del Negocio

- En el MBD, las reglas del negocio son expresiones que plantean dependencias entre datos → restricciones en las que ciertos valores en tuplas de una relación en una BD dependen de igualdades o desigualdades respecto a otros valores en la misma u otra relación o relaciones.
- Siempre que existan conjuntos de atributos compatibles y semánticamente vinculados es pertinente efectuar comparaciones entre ellos.
- En el modelo relacional las dependencias de datos primitivas son las dependencias funcionales y las dependencias de inclusión.
- A pesar de la sencillez del planteo, numerosas reglas del negocio pueden expresarse mediante dependencias de comparación de conjuntos de valores.

DCCV: Clases

- Clase de reglas del negocio de frecuente aparición en diferentes UdeD → basadas en dependencias de comparación de valores:
 - dependencias de inclusión
 - dependencias de exclusión
 - dependencias de igualdad
 - dependencias de superposición parcial
- Se pueden representar directamente en el MER (UML, ... modelo conceptual)?
- Es necesario y/o conveniente extender el MER (...) para capturar la semántica de estas reglas cuando sea imposible con las estructuras estandarizadas?
- Existe la posibilidad de efectuar transformaciones estructurales que proporcionen algún beneficio: simplicidad de la nueva representación?, nivel semántico del esquema?

DCCV: equivalencia.

- Análisis previo → identificar las que pueden plasmarse en un MER convencional directamente o mediante transformaciones.
- Si no es posible → formas de materialización alternativas equivalentes.
- La equivalencia de esquemas incluye varias perspectivas:
 - Estructural → se pueden soportar todas las restricciones estructurales impuestas sobre el modelo, es decir, la conversión es sin pérdida (lossless).
 - Semántica → todos y cada uno de los caminos formados por los vínculos semánticos (referencias) representan exactamente los conceptos modelados desde el UdeD → el esquema transformado preserva las dependencias funcionales → sumado a la propiedad de transformación sin pérdida garantiza que se preserve el conjunto específico de instancias durante la transformación o reconstrucción.
 - Operacional → habilidad del esquema transformado para manejar en forma equivalente las inserciones y supresiones de tuplas.

Dependencias de inclusión (DI).

- dependencias de inclusión + dependencias funcionales → pilares del modelo relacional → soportan la semántica esencial del modelo.
 - Permiten simbolizar vínculos semánticos significativos tales como relaciones y jerarquías.
 - Las restricciones de integridad referencial (RIR) son un caso particular de las DI y pueden ser definidas declarativamente vía la cláusula *foreign key* en cualquier BD de núcleo relacional
- ```
FOREIGN KEY (<referencing column list>) REFERENCES <referenced table name> [(<referenced column list>)
[MATCH <match type>]
[ON UPDATE <update referential action>]
[ON DELETE <delete referential action>]
```
- [ ( <referenced column list> ) ] → para referenciar columnas que no constituyen la clave primaria → se requiere la definición de una restricción de unicidad sobre esa columna.

---

---

---

---

---

---

---

---

### Restricciones de Integridad Referencial (RIR)

- Pueden ser especificadas con respecto a diferentes opciones de apareo (match type):
  - SIMPLE (la opción por defecto): para cada fila en la tabla referenciante, o bien la columna referenciante tiene al menos un valor nulo o su valor se corresponde con el valor de alguna fila en la tabla referenciada.
  - PARTIAL: para cada fila en la tabla referenciante el valor de cada columna de la clave extranjera es nulo, o bien hay al menos un valor no nulo que iguala a uno en la columna correspondiente en la tabla referenciada.
  - FULL: exige que para cada fila de la tabla referenciante, todos los valores de la clave extranjera sean nulos o bien que el conjunto de valores de la clave extranjera iguale el de las columnas correspondientes de alguna fila en la tabla referenciada.

---

---

---

---

---

---

---

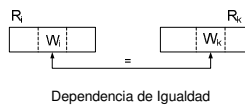
---

---

---

### Dependencias de igualdad (DIG)

- Una *dependencia de igualdad (DIG)*  $\equiv 2$  DI inversas.
- Existencia de un conjunto de atributos en una relación, cuyos valores deben ser los mismos que los correspondientes a un conjunto de atributos compatibles, en otra relación (o la misma)  $\rightarrow$  cada miembro del primer conjunto es a la vez miembro del segundo y viceversa.
- Una DIG denotada como  $R_i[W_i] = R_k[W_k]$  es equivalente a:  $R_k[W_k] \subseteq R_i[W_i] \wedge R_i[W_i] \subseteq R_k[W_k]$ , siendo  $W_k$  y  $W_i$  conjuntos de atributos compatibles.



Una dependencia de igualdad  $R_i[W_i] = R_k[W_k]$  se satisface si  $\forall t \in r_k$  existe una tupla  $t' \in r_i$  tal que  $t.W_k = t'.W_i$  y  $\forall t' \in r_i$  existe una tupla  $t \in r_k$  tal que  $t.W_k = t'.W_i$ .

---

---

---

---

---

---

---

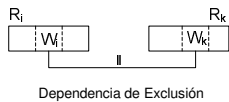
---

---

---

### Dependencias de Exclusión (DEX).

- Una *dependencia de exclusión (DEX)* se define como la existencia de atributos (simples o compuestos) en una relación, cuyos valores son mutuamente excluyentes respecto de un conjunto de valores de atributos compatibles (simples o compuestos respectivamente) en otra relación o en la misma. De esta manera ambos conjuntos de atributos no tienen miembros en común.



Formalmente una DEX se expresa  $R_i[W_i] \sqcap R_k[W_k]$  donde  $R_i$  y  $R_k$  son nombres de relaciones (posiblemente el mismo);  $W_i$  y  $W_k$  son conjuntos de atributos compatibles.  $R_i[W_i] \sqcap R_k[W_k]$  se satisface si no existe  $t \in r_k$  y  $t' \in r_i$ , tal que  $t.W_k = t'.W_i$ .

---

---

---

---

---

---

---

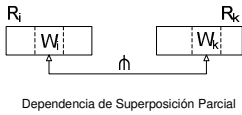
---

---

---

### Restricciones de superposición parcial (DSP)

- Las dependencias de superposición parcial (DSP) se denotan  $R_i[W_i] \hat{=} R_k[W_k]$
- Tienen un significado equivalente a:  $R_k \cdot W_k \not\subseteq R_i \cdot W_i \wedge R_i \cdot W_i \not\subseteq R_k \cdot W_k \wedge R_k \cdot W_i \not\subseteq R_i \cdot W_i$ , siendo  $W_k$  y  $W_i$  conjuntos de atributos compatibles.



$R_i[W_i] \hat{=} R_k[W_k]$  se satisface si existen  $t \in r_k$  y  $t' \in r_i$ , tal que  $t \cdot W_k = t' \cdot W_i$  y existe  $t \in r_k$  para la cual no existe  $t' \in r_i$  tal que  $t \cdot W_k = t' \cdot W_i$  y existe  $t' \in r_i$  para la cual no existe  $t \in r_k$ , tal que  $t \cdot W_k = t' \cdot W_i$ .

---

---

---

---

---

---

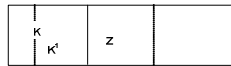
---

---

---

---

### Estructura Sintáctica de las Dependencias



Ubicaciones de un subconjunto de atributos con respecto a la clave primaria

- I)  $W=K$
- II)  $W=Z$
- III)  $W=K'$
- IV)  $W=K \cup Z$
- V)  $W=K' \cup Z$

- Pueden obtenerse 25 casos posibles de pares  $\langle R_i, W_i, R_k, W_k \rangle$ .
- Las DEX, DIG y DSP son simétricas, las DI y RIR son direccionales.

---

---

---

---

---

---

---

---

---

---

### Posibles conformaciones

| Término Derecho (W <sub>d</sub> ) / Término Izquierdo (W <sub>i</sub> )        | I) Clave (K <sub>d</sub> )                                          | II) No Clave (Z <sub>d</sub> )                                       | III) Parte de Clave (K <sub>d</sub> ' <sup>1</sup> )                                | IV) Clave + No Clave (K <sub>d</sub> ∪ Z <sub>d</sub> )                               | V) Parte de Clave + No Clave (K <sub>d</sub> ' <sup>1</sup> ∪ Z <sub>d</sub> )                       |
|--------------------------------------------------------------------------------|---------------------------------------------------------------------|----------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|
| I) Clave (K <sub>i</sub> )                                                     | 1. K <sub>i</sub> :: K <sub>d</sub>                                 | 6. K <sub>i</sub> :: Z <sub>d</sub>                                  | 11. K <sub>i</sub> :: K <sub>d</sub> ' <sup>1</sup>                                 | 16. K <sub>i</sub> :: K <sub>d</sub> ∪ Z <sub>d</sub>                                 | 21. K <sub>i</sub> :: K <sub>d</sub> ' <sup>1</sup> ∪ Z <sub>d</sub>                                 |
| II) No Clave (Z <sub>i</sub> )                                                 | 2. Z <sub>i</sub> :: K <sub>d</sub>                                 | 7. Z <sub>i</sub> :: Z <sub>d</sub>                                  | 12. Z <sub>i</sub> :: K <sub>d</sub> ' <sup>1</sup>                                 | 17. Z <sub>i</sub> :: K <sub>d</sub> ∪ Z <sub>d</sub>                                 | 22. Z <sub>i</sub> :: K <sub>d</sub> ' <sup>1</sup> ∪ Z <sub>d</sub>                                 |
| III) Parte de Clave (K <sub>i</sub> ' <sup>1</sup> )                           | 3. K <sub>i</sub> ' <sup>1</sup> :: K <sub>d</sub>                  | 8. K <sub>i</sub> ' <sup>1</sup> :: Z <sub>d</sub>                   | 13. K <sub>i</sub> ' <sup>1</sup> :: K <sub>d</sub> ' <sup>1</sup>                  | 18. K <sub>i</sub> ' <sup>1</sup> :: K <sub>d</sub> ∪ Z <sub>d</sub>                  | 23. K <sub>i</sub> ' <sup>1</sup> :: K <sub>d</sub> ' <sup>1</sup> ∪ Z <sub>d</sub>                  |
| IV) Clave + No Clave (K <sub>i</sub> ∪ Z <sub>i</sub> )                        | 4. K <sub>i</sub> ∪ Z <sub>i</sub> :: K <sub>d</sub>                | 9. K <sub>i</sub> ∪ Z <sub>i</sub> :: Z <sub>d</sub>                 | 14. K <sub>i</sub> ∪ Z <sub>i</sub> :: K <sub>d</sub> ' <sup>1</sup>                | 19. K <sub>i</sub> ∪ Z <sub>i</sub> :: K <sub>d</sub> ∪ Z <sub>d</sub>                | 24. K <sub>i</sub> ∪ Z <sub>i</sub> :: K <sub>d</sub> ' <sup>1</sup> ∪ Z <sub>d</sub>                |
| V) Parte de Clave + No Clave (K <sub>i</sub> ' <sup>1</sup> ∪ Z <sub>i</sub> ) | 5. K <sub>i</sub> ' <sup>1</sup> ∪ Z <sub>i</sub> :: K <sub>d</sub> | 10. K <sub>i</sub> ' <sup>1</sup> ∪ Z <sub>i</sub> :: Z <sub>d</sub> | 15. K <sub>i</sub> ' <sup>1</sup> ∪ Z <sub>i</sub> :: K <sub>d</sub> ' <sup>1</sup> | 20. K <sub>i</sub> ' <sup>1</sup> ∪ Z <sub>i</sub> :: K <sub>d</sub> ∪ Z <sub>d</sub> | 25. K <sub>i</sub> ' <sup>1</sup> ∪ Z <sub>i</sub> :: K <sub>d</sub> ' <sup>1</sup> ∪ Z <sub>d</sub> |

:: indica =, <<, ⊆, ||,  $\hat{=}$

---

---

---

---

---

---

---


---

---

---

Dependencias de inclusión (DI):

- Las DI → conceptos fundamentales en el diseño de BD.
- Las RIRs surgen naturalmente en el proceso de normalización u obtención de un esquema lógico a partir de un modelo semántico.
- Las DIs y algunas RIRs atípicas → originadas por otros factores.



---

---

---

---

---


---

---

---

DI - Factores "constructores"

- Refinamientos inapropiados del diseño lógico → ajustes ad-hoc del diseño lógico → práctica no recomendable pero efectuada corrientemente.
- Interpretación que el diseñador hace de ciertas reglas del negocio → representación en forma inadecuada. Ciertas reglas de la organización, en general enunciadas textualmente en forma sencilla y con ambigüedades, pueden plasmarse directamente en el esquema lógico mediante inclusiones, sin necesidad de conversión alguna.
- Variados casos de desnormalización por razones prácticas → la desnormalización "desplaza" el objetivo de una referencia y lo ubica en el esquema de otra tabla, en una posición cualquiera respecto de la clave primaria de la tabla receptora dependiendo de la estructura del vínculo original.



---

---

---

---

---


---

---

---

DI - Factores "constructores"

- Conversión a un modelo binario de esquemas con relaciones de aridad mayor que 2 con o sin restricciones adicionales impuestas → las dependencias surgen por la utilización de patrones de transformación sobre esquemas conceptuales correctos planteados según modelos no binarios → convertirlos a binarios, porque todas las herramientas CASE de soporte al diseño conceptual siguen este modelo → necesidad de adicionar restricciones adicionales a las que pueden ser derivadas algorítmicamente, con el objeto de asegurar la equivalencia semántica y operacional.
- Problemas de integración de BD → tablas similares provenientes de diferentes BD fuente tienen diferentes restricciones referenciales especificadas. Además, cada DBMS ofrece sus propias facilidades de especificación → como producto de la integración, pueden violarse algunas reglas referenciales, o bien pueden distorsionarse o ignorarse otras.



---

---

---

---

---

---

---

---

### DI - Diseños inapropiados

- Modelado conceptual → los objetos y las conexiones entre ellos se deberían representar apropiadamente.
- La evolución natural del sistema modelado promueve modificaciones del esquema → frecuentemente son realizadas ad-hoc, sin respetar los lineamientos de la metodología escogida, o sin seguir un procedimiento ordenado.
- Algunas dependencias que fueron correctamente detectadas o previstas en estadios tempranos del diseño, luego fueron incorrectamente representadas.
- Actividades que generalmente quedan indebidamente documentadas → la comprensión y/o exactitud de los datos se va perdiendo gradualmente, y el sistema se vuelve poco flexible, difícil de mantener e inadecuado para las nuevas aplicaciones.

---

---

---

---

---

---

---

---

### DI - Diseños inapropiados

- Cuando los sistemas no cumplimentan los requisitos de calidad esperados → aplicación de estrategias de reingeniería propende a mejorar la situación.
  - No necesariamente debe ser integral; en algunos casos es suficiente una reingeniería local o parcial → un esquema de BD con las deficiencias mencionadas puede ser restaurado y llevado a un estado bien definido, a partir del cual se pueda trabajar con las herramientas, métodos y heurísticas convenientes.
- El estudio de los orígenes de las *DI*s y de ciertos casos atípicos de *RIRs*, juntamente con la investigación de estrategias para su remoción y/o tratamiento → conversión de esquemas incorrectos y difíciles de mantener en esquemas normalizados.

---

---

---

---

---

---

---

---

### DI - Desnormalización

- Un diseño de una BD adhiriendo estrictamente a una metodología de diseño sólo produce relaciones entre entidades propiamente modeladas, las que son derivadas en relaciones designativas o asociativas en el modelo lógico.
- Sin embargo es frecuente que los usuarios especifiquen el esquema dentro del modelo relacional convencional → usando tablas que no se encuentran en la forma normal más apropiada (al menos 3FN, idealmente FNBC) sino en alguna forma normal previa.
  - La desnormalización unifica objetos eliminando relaciones designativas, o fundiendo relaciones asociativas con uno o más de sus objetos asociados → "desplazamiento" del objetivo de la referencia ubicándolo en el esquema de otra tabla !!
  - Este escenario es muy común y puede ocurrir debido a que ciertos requerimientos de las aplicaciones (por ejemplo la necesidad de reducir el número de ensambles), se imponen a las ventajas de una normalización apropiada.

---

---

---

---

---

---

---

---

### DI - Conversión de relaciones n-arias con o sin binarias relacionadas semánticamente

- MER → herramienta estándar de facto para el diseño de BDR.
- Representar correctamente estructuras conceptuales complejas → facilidades para incorporar mayor contenido semántico.
  - Relaciones que en el MR se plantean entre tres o más entidades, al ser tratadas con las herramientas CASE deben ser transformadas.
  - Esquema binario resultante → estructural, semántica y operacionalmente equivalente al original.
  - Transformación no siempre directa → + restricciones además de las dep. funcionales y restricciones de integridad referencial derivadas.
- Relaciones ternarias o de mayor orden → → se transforman en binarias relacionadas siguiendo diferentes estrategias: 1) agregaciones; 2) 'reificación' u 'objetificación' mediante entidades denominadas 'asociativas', 'intersección' o 'gerundio' y 3) transformaciones guiadas por las dependencias funcionales del esquema original

---

---

---

---

---

---

---

---

---

---

### DI - Integración

- Integración de BD representa un escenario común en el cual:
  - múltiples BD 'fuente' tienen diferentes restricciones de integridad,
  - cada DBMS provee diferentes mecanismos, reglas y esquema de ejecución para forzar la integridad.
- Las BD originales pueden violar la integridad global.
  - la integración puede revelar problemas de integridad adicionales.

---

---

---

---

---

---

---

---

---

---

### Restricciones de Integridad Referencial (RIR)

- RIRs de tipos I, II y III → provenientes de un modelo de datos o normalización.
- Tipo I permite la representación de relaciones de tipo-subtipo
- Tipo II permiten abstraer relaciones designativas de cardinalidad 1:1, N:1 o n-arias con al menos una cardinalidad 1
- Tipo III aparecen en relaciones asociativas N:N, n-arias en general y las correspondientes a los vínculos entidad fuerte-entidad débil.
- Tipos IV y V son atípicas y requieren un análisis diferente.

---

---

---

---

---

---

---

---

---

---

### RIR - Análisis del caso IV

- $R_i[K_i \cup Z_i] \ll R_d[K_d]$  (celda 4)
- EJEMPLO:

$R_i$ : INSCR\_EN\_CONCURSO (#Estudiante, #Curso, FechaInscrip)  
 $R_d$ : ESTUDIANTE-APROBADO (#Estudiante, #Curso, NotaFinal).

- $R_i$ : alumnos aspirantes a un cargo de auxiliar de cátedra en un curso
- $R_d$ : notas de examen final para alumnos aprobados en materias de una carrera.

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

DI =  $\{R_i[\#Estudiante, \#Curso] \ll R_d[\#Estudiante, \#Curso]\}$

22

---

---

---

---

---

---

---

---

---

---

---

---

### RIR - Análisis del caso IV

- #Curso tiene una semántica diferente en una y otra tabla  $\rightarrow$  los pares #Estudiante, #Curso también tienen un significado diferente en ambas tablas.
- Si existen diferentes dependencias funcionales involucrando los mismos atributos  $\rightarrow$  diferencias semánticas.
- La DI adicionada  $\rightarrow$  *regla del negocio*: "Los estudiantes pueden inscribirse en un concurso para desempeñarse como auxiliares, sólo si han aprobado el examen de esa materia".
- Analizando la estructura de las tablas derecha e izquierda:
  - La tabla derecha es una relación derivada de una relación N:N entre ESTUDIANTE y CURSO.
  - Un Diagrama Entidad Relación (DER) podría ser el de la Figura siguiente.
  - $R_2$  es la relación designativa que aparentemente representaría la RIR bajo análisis.

23

---

---

---

---

---

---

---

---

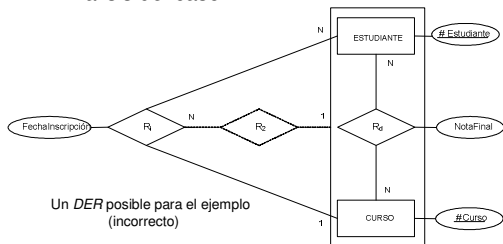
---

---

---

---

### RIR - Análisis del caso IV



- La semántica de la regla no queda expresada correctamente  $\rightarrow R_2$  al ser derivada en el esquema relacional quedaría definida en función de la clave de  $R_1$  (que en realidad puede desplazarse a ESTUDIANTE).
- No puede obtenerse la RIR a partir de la derivación DER  $\rightarrow$  esquema relacional.

24

---

---

---

---

---

---

---

---

---

---

---

---



**RIR - Análisis del caso IV**

- El *DER* presenta ambigüedades. Formalmente, paso a paso:

ESTUDIANTE (#Estudiante, ...) (1)  
CURSO (#Curso, ...) (2)  
 $R_d$  (#Estudiante, #Curso, NotaFinal) (3)

(1), (2) y (3) → derivación inicial de las respectivas entidades y relación. Incorporando a (1) la derivación de la relación  $R_1$  (N:1) → (4)

ESTUDIANTE (#Estudiante, ..., #Curso) (4)

La derivación de  $R_2$  y su incorporación en (4) → (5).  
**ESTUDIANTE(#Estudiante, ..., #Curso, #EstudAsp, #CursoA) (5)**  
(2), (3) y (5) son las relaciones resultantes. #EstudAsp y #CursoA renombran a #Estudiante y #Curso para  $R_2$ .

---

---

---

---

---

---

---

---

---

---

---

---

**RIR - Análisis del caso IV**

- Las *RIRs* correspondientes son:

$R_d$  [#Estudiante] << ESTUDIANTE[#Estudiante]  
 $R_d$  [#Curso] << CURSO[#Curso]  
ESTUDIANTE[#Curso] << CURSO[#Curso]  
ESTUDIANTE [#EstudAsp, #CursoA] <<  $R_d$  [#Estudiante, #Curso]

- Este esquema no representa adecuadamente la semántica de la regla del negocio !!!
- Es necesario que  $\forall t \in \text{ESTUDIANTE}: t.\#Estudiante = t.\#EstudAsp$  y  $t.\#Curso = t.\#CursoA$

No se puede representar en el *DER* → (5) se convierte en ESTUDIANTE (#Estudiante, ..., #Curso) y la *RIR* es ESTUDIANTE[#Estudiante, #Curso] <<  $R_d$  [#Estudiante, #Curso]  
Recordar que  $R_1 \equiv \text{ESTUDIANTE}$

---

---

---

---

---

---

---

---

---

---

---

---

**RIR - Análisis del caso IV**

- Otra perspectiva: Dado que #Curso y #Estudiante en  $R_1$  y  $R_d$  han sido considerados en diferentes contextos, observando la existencia de atributos inaplicables → detección de una relación de tipo-subtipo
- La relación punteada nuevamente parece representar la *regla del negocio*.

*DER* aparentemente más cercano a la semántica del ejemplo en el MR (incorrecto).

---

---

---

---

---

---

---

---

---

---

---

---

**RIR - Análisis del caso IV**

CURSO (#Curso, ...) (6)

ESTUDIANTE (#Estudiante, ...) (7)

$R_d$  (#Curso, #Estudiante, NotaFinal) (8)

CURSO\_CON\_POSICION\_VACANTE (#Curso, ..., #Estudiante, #CursoP, #EstudianteA) (9)

ESTUDIANTE\_ASPIRANTE (#EstudAsp, FechaInscripción, #CursoPV, #Curso, #Estudiante) (10)

Las RIRs son:

$R_d$  [#Curso] << CURSO [#Curso]

$R_d$  [#Estudiante] << ESTUDIANTE [#Estudiante]

ESTUDIANTE\_ASPIRANTE [#EstudAsp] << ESTUDIANTE [#Estudiante]

ESTUDIANTE\_ASPIRANTE [#Curso, #Estudiante] <<  $R_d$  [#Curso, #Estudiante].

---

---

---

---

---

---

---

---

---

---

---

---

**RIR - Análisis del caso IV**

Nuevamente debe satisfacerse

$\forall t \in \text{ESTUDIANTE\_ASPIRANTE} : t.\#CursoPV = t.\#Curso \text{ y } t.\#EstudAsp = t.\#Estudiante,$

aspecto no representable en el DER. Esto permite resumir (9) en ESTUDIANTE\_ASPIRANTE(#Estudiante, FechaInscripcion, #Curso) y transformar la RIR correspondiente en:

ESTUDIANTE\_ASPIRANTE[#Curso, #Estudiante] <<  $R_d$  [#Curso, #Estudiante].

$R_d$  está empotrada en ESTUDIANTE\_ASPIRANTE.

---

---

---

---

---

---

---

---

---

---

---

---

**RIR - Análisis del caso IV**

- Obtención directa de un DER para estos casos: no es posible  $\rightarrow$  se requiere establecer una restricción de igualdad entre valores de atributos.
- Sin embargo la calidad del esquema ha sido mejorada porque:
  - se han representado correctamente los diferentes roles de las entidades en las dos relaciones;
  - se ha plasmado correctamente la relación  $R_d$ ;
  - se han identificado las correspondientes relaciones tipo-subtipo, eliminando atributos y relaciones inaplicables.

---

---

---

---

---

---

---

---

---

---

---

---

### RIR - Análisis del caso IV, representación

- En SQL → se declaran habitualmente (aunque incorrectamente) como chequeos asociados a una tabla (en este caso R<sub>i</sub>) de la siguiente forma:
- CHECK ((K<sub>i</sub>, Z<sub>i</sub>) IN (SELECT K<sub>d</sub><sup>1</sup>, K<sub>d</sub><sup>2</sup> FROM R<sub>d</sub>))
- Para que sea correcto → se requiere que el CHECK tenga la semántica de una aserción, es decir de una regla de integridad general → problemas de chequeos con subconsultas pueden producir inconsistencias
- En gral: CHECK(<referencing column> IN (SELECT <referenced column> FROM <referenced table>))

---

---

---

---

---

---

---

---

### RIR - Análisis del caso IV

*La tabla derecha es una relación derivada de una débil*

Sea el siguiente fragmento del esquema conceptual de una BD (claves subrayadas):

PAÍS (IdPaís, NombrePaís, Capital, Población) (regular) (1)  
CIUDAD (NombreCiudad, País, Intendente) (débil) (2)

Las RIRs son:

CIUDAD[País] << PAÍS[IdPaís] y

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

La última RIR es de tipo IV pero podría ser establecida como una de tipo II.

---

---

---

---

---

---

---

---

### RIR - Análisis del caso IV

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 se debe cumplir  $\forall t \in \text{PAÍS}: t.\text{IdPaís} = t.\text{IdPaís2}$ .

No hay otra forma de representación que supere ésta. A modo de justificación, y suponiendo que los nombres de las ciudades capitales se pueden repetir (tal como indica (1)), considérense las siguientes alternativas:

1) CAPITAL podría configurarse como una clase de entidades, con relación 1:N con PAÍS, reiterando (1). Esta relación constituye una agregación, que está en relación 1:1 con CIUDAD. Esta forma de representación presenta similares ambigüedades que el caso anterior

---

---

---

---

---

---

---

---

### RIR - Análisis del caso IV

2) CAPITAL es una ciudad especial, por ende sería correcto expresarla en un subtipo de CIUDAD. CIUDAD es débil y hereda su clave a CAPITAL. De esta manera el esquema lógico resulta:

PAÍS (IdPaís, NombrePaís, Capital, Población)

CIUDAD (NombreCiudad, País, Intendente)

CAPITAL (NbreCiudadCapital, País) (3)

Con las rirs: CIUDAD[País] << PAÍS[IdPaís]; CAPITAL [País, NbreCiudadCapital] << CIUDAD[País, NombreCiudad] . Teniendo en cuenta que PAÍS: IdPaís → Capital , resulta que (3) no está normalizada!!.

---

---

---

---

---

---

---

---

### Análisis de las DI.

- Las dependencias de inclusión no basadas en claves → una relación oculta más de un objeto → la referencia al objeto empotrado genera una DI.
- Se pone énfasis en el estudio de los términos derechos para:
  - encontrar las razones que justifican el diseño de estas restricciones
  - proponer formas alternativas de especificación mediante la identificación de objetos empotrados y reglas del negocio ocultas.
- Se presenta una heurística que permite un rediseño parcial del esquema y sus restricciones de referencia, mejorando la calidad respecto de sus aspectos semánticos y de los relativos a facilitar su interpretación por parte del usuario.

---

---

---

---

---

---

---

---

### Análisis de las DI. Término derecho tipo I: $R_1[W_i] \ll R_d[K_d]$

Instancias 1 a 5 de la Tabla (RIRs sintácticamente incluidas en este caso).

Sea el siguiente segmento de un esquema relacional:

(R<sub>1</sub>) ESTUDIANTE (Id-estud, Nombre-estudiante, Carrera, Director-de-tesis).

(R<sub>d</sub>) INSTRUCTOR (Id-estud, Título)

La aparente RIR es INSTRUCTOR[Id-estud] << ESTUDIANTE[Id-estud]

Supóngase que 'Director-de-tesis' es un atributo que admite nulos y en realidad es inaplicable (Codd, 1990), ya que es "no nulo" sólo cuando el estudiante es graduado. En este caso el diseño correcto del conjunto de relaciones debió ser:

(R<sub>1</sub><sup>1</sup>) ESTUDIANTE (Id-estud, Nombre-estudiante, Carrera)

(R<sub>d</sub><sup>2</sup>) ESTUDIANTE-GRADUADO (Id-estud, Director-de-tesis)

(R<sub>d</sub>) INSTRUCTOR (Id-estud, Título)

---

---

---

---

---

---

---

---



**Análisis de las DI. Término derecho tipo II:  $R_i[W_i] \subseteq R_d[Z_d]$**

Casos 6 a 10 en la Tabla.

Sean PRESUPUESTO ( $R_i$ ) con datos correspondientes a subsidios y PROYECTO ( $R_d$ ) con datos relativos a proyectos en desarrollo en los departamentos de una organización.

PRESUPUESTO ( $\#Expediente$ , NbreDepto, Monto) y

PROYECTO ( $\#Proyecto$ , NbreProyecto, NbreDepto, FechaComienzo)

Sea la DI PRESUPUESTO [NbreDepto]  $\subseteq$  PROYECTO [NbreDepto].

Objeto omitido  $\rightarrow$  DEPARTAMENTO.

La DI puede ser interpretada como "sólo aquellos departamentos teniendo proyectos en marcha pueden tener presupuesto asignado".

---

---

---

---

---

---

---

---

**Análisis de las DI. Término derecho tipo II:  $R_i[W_i] \subseteq R_d[Z_d]$**

En ese caso el diseño lógico correcto debe ser:

$R_d^1$ : PROYECTO ( $\#Proyecto$ , NbreProyecto, NbreDepto, FechaComienzo)

$R_i$ : PRESUPUESTO ( $\#Expediente$ , NbreDepto, Monto) y  
 $R_d^2$ : DEPARTAMENTO (NbreDepto).

El conjunto DI resultante es:

PRESUPUESTO [NbreDepto]  $\ll$  DEPARTAMENTO [NbreDepto]

y PROYECTO [NbreDepto]  $\ll$  DEPARTAMENTO [NbreDepto].

Y la restricción entre comillas es la siguiente DI remanente (DIR) indicando la regla del negocio

DEPARTAMENTO [NbreDepto]  $\subseteq$  PROYECTO [NbreDepto].

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

---

---

---

---

---

---

---

---

**Análisis de las DI. Término derecho tipo II:  $R_i[W_i] \subseteq R_d[Z_d]$**

Por el contrario, si se intenta hacer referencia a cualquier Departamento, el esquema rediseñado sería el mismo y la DI original se reemplazaría con las siguientes RIRs:

PRESUPUESTO [NbreDepto]  $\ll$  DEPARTAMENTO [NbreDepto]

y PROYECTO [NbreDepto]  $\ll$  DEPARTAMENTO [NbreDepto].

---

---

---

---

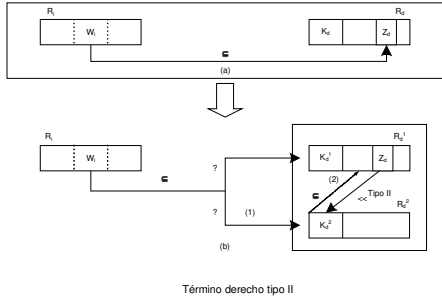
---

---

---

---

Análisis de las *DI*. Término derecho tipo II:  $R_i[W_i] \subseteq R_d[Z_d]$




---

---

---

---

---

---

---

---

Análisis de las *DI*. Término derecho tipo IV:  $R_i[W_i] \subseteq R_d[K_d \cup Z_d]$

Casos 16 a 20 en la Tabla.

No es posible conocer cuál de las dos relaciones colapsadas en el término derecho es el objetivo de la referencia.

Si se requiere referenciar  $R_d^2$ , los conjuntos correctos de relaciones y restricciones deben ser reformulados en forma análoga a los casos anteriores.

Si  $W_i$  es una clave alternativa, se puede percibir una sobrecarga semántica

Si  $W_i$  es una clave primaria, esto indica que  $R_i:W_i \rightarrow K_i$ , constituyendo un diccionario también en  $R_i$ .

Los restantes casos tienen una simplificación particular.

---

---

---

---

---

---

---

---

Análisis de las *DI*. Término derecho tipo IV:  $R_i[W_i] \subseteq R_d[K_d \cup Z_d]$

Sean las relaciones

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

y la *DI* DIRECCIÓN [IdDirector, IdDepto]  $\subseteq$  NÓMINA [IdProfesor, IdDepto]  $\rightarrow$

el director de un departamento debe ser un integrante de la nómina de profesores de ese mismo departamento.

A partir de la *DI* y de la dep. funcional NÓMINA: IdProfesor  $\rightarrow$  IdDepto puede inferirse DIRECCIÓN: IdDirector  $\rightarrow$  IdDepto.

Esto ilustra un error de diseño bastante frecuente.

Consecuencias:

- la interacción de dependencias funcionales y de inclusión puede dar lugar a restricciones de integridad derivadas que el diseñador no tenía previstas;
- 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.

---

---

---

---

---

---

---

---





### Conversión de Esquemas con *DIs*.

- Sean  $DI = \{ \dots; R_j[W_j] \subseteq R_j[W_d]; \dots \}$   $j = 1, 2, \dots$  y  $DF = \{ \dots; R_i; K_i \rightarrow \text{atr}(R_i) - K_i; R_j; K_j \rightarrow \text{atr}(R_j) - K_j; R_d; W_d \rightarrow Z_d^2; \dots \}$ .
- 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 debe ser resuelto con la regla de "pullback".
- Sea el esquema normalizado  $R^n = \langle R^n, D^n \rangle$ , con  $D^n = \{ DF^n, DI^n \}$ .
- Una vez que las entidades ocultas se han detectado y sus respectivas clases se han identificado, los siguientes pasos conducen el proceso de reingeniería parcial:

---

---

---

---

---

---

---

---

---

---

---

---

### Conversión de Esquemas con *DIs*.

- Input:** Un esquema relacional  $R = \langle R, D \rangle$  con entidades y relaciones ocultas;
- Output:** Un esquema relacional  $R^n = \langle R^n, D^n \rangle$  normalizado;
- For** Cada entidad oculta detectada según el análisis del término derecho  $R_d[W_d]$  de *dIs* en DI, crear e incorporar a  $R^n$  una relación NUEVA con:

```

KNUEVA = Wd;
If Existen dIs con la estructura Rd; Wd → Zd2 /* Hay atributos descriptivos que corresponden a la entidad oculta en Rd */
Then Transferirlos como atributos descriptivos en NUEVA; /* atr(NUEVA) = Wd, Zd2 */
Eliminarlos de Rd, convirtiendo Rd en Rd' /* atr(Rd) = Kd, Zd1 */
EndIf;

```

---

---

---

---

---

---

---

---

---

---

---

---

### Conversión de Esquemas con *DIs*.

- Caso II, III o V:**
  - For** Cada  $R_j[W_j] \subseteq R_d[W_d]$   $j = 1, 2, \dots$  /\* el término izquierdo es el conjunto de atributos que identifican la entidad omitida en NUEVA \*/
    - Redefinir la DI como  $R_j[W_j] \ll \text{NUEVA}[K_{\text{NUEVA}}]$  --- (2)
    - EndFor**
    - Incluir  $R_j[W_j] \ll \text{NUEVA}[K_{\text{NUEVA}}]$  en  $DI^n$ ; --- (3)
    - If** existiera una regla del negocio oculta
      - Then** Agregar una  $DI^n$   $\text{NUEVA}[K_{\text{NUEVA}}] \subseteq R_j[W_j]$  a  $DI^n$  --- (4)
      - EndIf**

---

---

---

---

---

---

---

---

---

---

---

---

## Conversión de Esquemas con *DIs*.

### Caso I:

For Cada *di* definida  $R_i[W_i] \subseteq R_d[W_d], i = 1, 2, \dots, n$  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  $R_i[W_i] \ll \text{NUEVA}[K_{\text{NUEVA}}]$  ---- (2)  
 Else Incluir  $R_i[W_i] \ll R_d[K_d]$  en  $\text{DI}^n$   
 EndIf.

If NUEVA y  $R_i$  están vinculadas por una relación 1:1 mandatoria  
 Then Incluir  $R_i[W_i] \ll \text{NUEVA}[K_{\text{NUEVA}}]$  en  $\text{DI}^n$  ---- (3)  
 EndIf.

Agregar una *DIR*  $\text{NUEVA}[K_{\text{NUEVA}}] \subseteq R_d[W_d]$  a  $\text{DI}^n$  ---- (4)  
 EndFor;  
 End.

---

---

---

---

---

---

---

---

## Conversión de Esquemas con *DIs*.

La aplicación de estos pasos resulta en:

- $\text{R}^n = R \cup \{\text{NUEVA}; R_d\} - \{R_d\}$ .
  - Para los casos II, III y V,  $\text{DI}^n = \text{DI} \cup \{R_d[W_d] \ll \text{NUEVA}[K_{\text{NUEVA}}]\} \cup \{R_i[W_i] \ll \text{NUEVA}[K_{\text{NUEVA}}]\}_{i=1,2,\dots} - \{R_i[W_i] \subseteq R_d[W_d]\}_{i=1,2,\dots}$ .
  - Para el caso indicado en (4)  $\text{DI}^n = \text{DI} \cup \{R_d[W_d] \ll \text{NUEVA}[K_{\text{NUEVA}}]; \text{NUEVA}[K_{\text{NUEVA}}] \subseteq R_d[W_d]\} \cup \{R_i[W_i] \ll \text{NUEVA}[K_{\text{NUEVA}}]\}_{i=1,2,\dots} - \{R_i[W_i] \subseteq R_d[W_d]\}_{i=1,2,\dots}$ .
- $\text{DF}^n = \text{DF} \cup \{\text{NUEVA}:K_{\text{NUEVA}} \rightarrow Z_d^2\} - \{R_d:K_d \rightarrow Z_d^2, R_d:W_d \rightarrow Z_d^2\}$ .
- Para el caso I,  $\text{DI}^n$  y  $\text{DF}^n$  pueden ser obtenidos de manera similar.

---

---

---

---

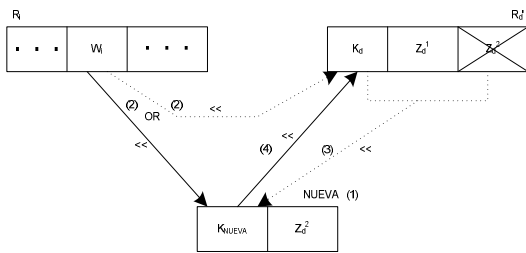
---

---

---

---

## Conversión Caso I




---

---

---

---

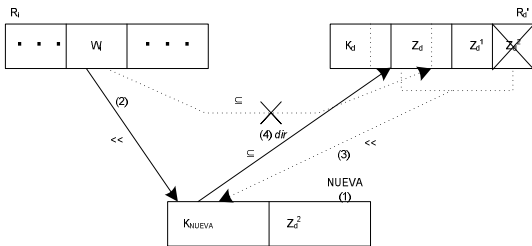
---

---

---

---

### Conversión Casos II, III, V




---

---

---

---

---

---

---

---

### Dependencias de igualdad: un ejemplo.

Términos de tipo II y IV:  $R_i[Z_i] = R_k[K_k, Z_k]$

Sea una DIG estableciendo que los valores de los atributos Id\_P (identificador de proyecto), Id\_D (identificador de departamento) en la relación CURSO deben ser los mismos que los correspondientes al conjunto de atributos compatibles Id\_P, Id\_D, en la relación PROFESOR.

El primer paso consiste en determinar si existe alguna transformación de las D/s que permita obtener un patrón simplificado representable mediante los mecanismos de abstracción estándar del MER, y si existe algún vínculo semántico redundante.

**Regla 1:** Un profesor (Id\_P) puede dictar un curso (Id\_C) sólo si pertenece al mismo departamento (Id\_D) que ofrece el curso.

**Regla 2:** Un profesor (Id\_P) pertenece a un departamento (Id\_D) sólo si está dictando al menos un curso (Id\_C) para ese departamento.

---

---

---

---

---

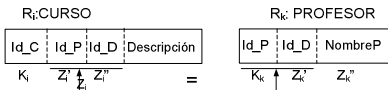
---

---

---

### Dependencias de igualdad: un ejemplo.

Términos de tipo II y IV:  $R_i[Z_i] = R_k[K_k, Z_k]$



**CURSO[id\_P, id\_D] = PROFESOR[id\_P, id\_D]**

Dependencias de inclusión

Tipo IV: **CURSO[id\_P, id\_D] ⊆ PROFESOR[id\_P, id\_D]** ( $di_1$ )

Tipo II: **PROFESOR[id\_P, id\_D] ⊆ CURSO[id\_P, id\_D]** ( $di_2$ )

---

---

---

---

---

---

---

---

### Dependencias de igualdad: un ejemplo.

Si se resuelve la  $D_1$  por la *regla de pullback* se tiene que  $\text{PROFESOR.Id}_P \rightarrow \text{PROFESOR.Id}_D$ , entonces  $\text{CURSO.Id}_C \rightarrow \text{CURSO.Id}_D$  es una dependencia transitiva. Así en CURSO sólo quedaría  $\text{CURSO.Id}_C \rightarrow \text{CURSO.Id}_P$  y  $D_1$  se transforma en la *RIR*:

$\text{CURSO}[\text{Id}_P] \ll \text{PROFESOR}[\text{Id}_P]$

Esta transformación reduce  $D_2$  a  $\text{PROFESOR}[\text{Id}_P] \subseteq \text{CURSO}[\text{Id}_P]$ , que continúa siendo de Tipo II pero con su término derecho diferente del original.

- El análisis continúa analizando si tiene sentido aplicar en este punto la heurística de transformación.
- La aplicación de la transformación con el objetivo de simplificar el esquema produce el efecto deseado ya que persiste una  $D_1$  de tipo II. Aparentemente se ha enriquecido el esquema al poner en evidencia NUEVA, pero en realidad es una proyección de las columnas correspondientes a la clave de PROFESOR.

---

---

---

---

---

---

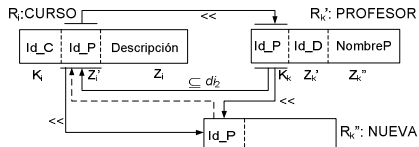
---

---

---

---

### Dependencias de igualdad: un ejemplo.




---

---

---

---

---

---

---

---

---

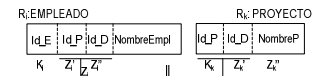
---

### Dependencias de Exclusión: un ejemplo.

Ejemplo 2. De – Términos de tipo II y IV:  $R_1[Z_1] \sqcap R_2[K_2, Z_2]$

Un empleado ( $\text{Id}_E$ ) no puede evaluar proyectos ( $\text{Id}_P$ ) del departamento ( $\text{Id}_D$ ) para el cual trabaja.

Regla en lenguaje natural



Dependencia de exclusión  
 $\text{EMPLEADO}[\text{Id}_P, \text{Id}_D] \parallel \text{PROYECTO}[\text{Id}_P, \text{Id}_D]$   
 Regla expresada como *de*

---

---

---

---

---

---

---

---

---

---

### Dependencias de Exclusión: un ejemplo.

En este caso la relación entre PROYECTO y DEPARTAMENTO no puede ser transformada con los mecanismos conocidos basados en *Ds*, ya que una *DE* no es simplemente la negación de una *DI*.

Este tipo de dependencias no es expresable en el *MER* convencional.

Podría especificarse mediante sendas restricciones de tabla:

CHECK ((Empleado.Id\_P, Empleado.Id\_D) NOT IN (Select Id\_P, Id\_D FROM Proyecto))

y

CHECK ((Proyecto.Id\_P, Proyecto.Id\_D) NOT IN (Select Id\_P, Id\_D FROM Empleado))

61

---

---

---

---

---

---

---

---

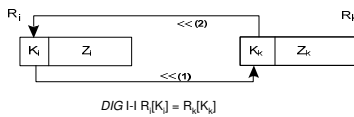
---

---

### Dependencias de igualdad: estudio de casos

Caso 1.- Es trivial, aunque es una estructura poco común.

- Ambas restricciones son *RIFs*.
- En un *DER*, una de las *RIFs* podría ser representada mediante una jerarquía, no así la inversa.
- Clase de *DIGs* que aparece cuando se proyectan tablas en forma vertical, por requerimientos de seguridad o desempeño.
- La *DIG* se genera por la necesidad de mantener la igualdad entre los dominios activos de la clave, presente en ambas proyecciones. Operativamente presenta la problemática de los ciclos referenciales.
- No hay transformación que pueda mejorar esta situación.



62

---

---

---

---

---

---

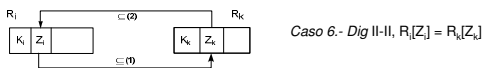
---

---

---

---

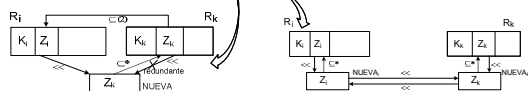
### Dependencias de igualdad: estudio de casos



A partir de esta estructura pueden aplicarse dos transformaciones alternativas.

- conservar la *di* (2) tal como ha sido definida aplicando la heurística a la *di* (1) (o su caso simétrico):

- aplicar la heurística a ambas *di*



*Dig* II-II convertida. \* necesaria para preservar la semántica original

*Dig* II-II convertida. \* necesarias para preservar la semántica original

63

---

---

---

---

---

---

---

---

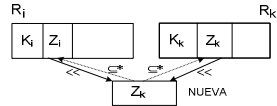
---

---

### Dependencias de igualdad: estudio de casos

- La aplicación de la transformación no garantiza que la igualdad original se preserve ya que  $Z_i$  y  $Z_k$ , aunque referencien conjuntos idénticos, no necesariamente tienen el mismo dominio activo en todo tiempo, en consecuencia ambas *di* señaladas con \* son necesarias. No tiene sentido tener ambas tablas NUEVA y NUEVA<sub>k</sub>, debería simplificarse de la siguiente forma:
- Nuevamente, con la especificación de ambas *di* remanentes se consigue la semántica original.

*Dig* II-II convertida y simplificada.




---

---

---

---

---

---

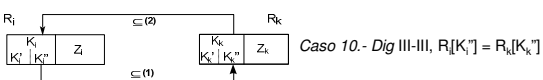
---

---

---

---

### Dependencias de igualdad: estudio de casos



- $R_i$  o bien  $R_k$  podrían ser relaciones N:N o entidades débiles.
- De los dos primeros casos puede hacerse el siguiente análisis:
- $R_i$  y  $R_k$  son relaciones binarias N:N.
- Traduciendo el esquema de la Figura a un DER, se obtendría:

---

---

---

---

---

---

---

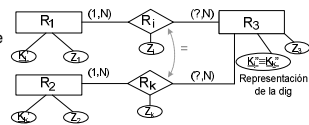
---

---

---

### Dependencias de igualdad: estudio de casos

- La equivalencia entre  $R_3[K_i]$  y  $R_3[K_k]$ .
- No es posible representar gráficamente la *dig* (superpuesta en la gráfica con líneas punteadas). La obligatoriedad de  $R_3$  en el vínculo con  $R_1$ , y en el correspondiente a  $R_2$  representa las *digs*:
- $R_1[K_i] = R_3[K_i]$
- $R_k[K_k] = R_3[K_k]$ , pero como
- $R_3[K_i] \equiv R_3[K_k]$ , de allí surge
- $R_1[K_i] = R_k[K_k]$




---

---

---

---

---

---

---

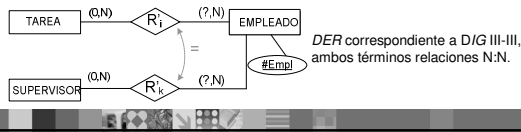
---

---

---

### Dependencias de igualdad: estudio de casos

- Sin embargo esta es una dependencia más restrictiva que la original, pues  $R_i, K_i^+$  podría tener los mismos valores que  $R_k, K_k^+$  sin ser necesariamente equivalentes a  $R_2, K_2^+$  (o a  $R_3, K_3^+$ )  $\rightarrow$  no necesariamente el vínculo tiene que ser obligatorio en el sentido  $R_i/R_2$
- Las cardinalidades mínimas podrían ser 0 (cero) en los extremos  $R_1$  y  $R_2$ ,  $\rightarrow$  aquéllos ejemplares de  $R_3$  que no tengan pareja en  $R_1$ , tampoco deben tenerla en  $R_2$ . Ejemplo: "aquéllos empleados que realizan tareas, necesariamente tienen supervisores y quienes tienen supervisores, necesariamente están desarrollando tareas"  $\rightarrow$  no hay supervisores de empleados que no desarrollan tareas ni empleados desarrollando tareas sin supervisión. El DER correspondiente a esta situación:




---

---

---

---

---

---

---

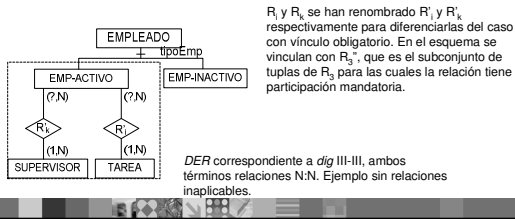
---

---

---

### Dependencias de igualdad: estudio de casos

- Para garantizar que sólo los ejemplares de #Empl en  $R_i$  aparecen en  $R_k$  y viceversa (que es la representación de la *dig*  $R_i[\#Empl] = R_k[\#Empl]$  que describe la regla del negocio) podría emplearse una extensión del concepto de *inaplicabilidad* (Codd, 1990), ya que debe darse en forma simultánea para las dos relaciones.
- De esta forma,




---

---

---

---

---

---

---

---

---

---

### DIGs tácitas en transformaciones de esquemas equivalentes

Sea la relación ternaria N:1:1 de la Figura.

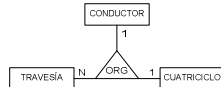
Las relaciones derivadas de este esquema son (claves primarias subrayadas):

TRAVESÍA (#t, ...,) CUATRICICLO (#c, ...)

ORG (#c, #t, #co)

y las dependencias funcionales primarias en ORG (las correspondientes a las entidades son obvias):

ORG: #c #t  $\rightarrow$  #co      ORG: #t #co  $\rightarrow$  #c



Las *d/s* son: ORG[#c]  $\subseteq$  CUATRICICLO[#c]; ORG[#co]  $\subseteq$  CONDUCTOR[#co] y ORG[#t]  $\subseteq$  TRAVESÍA[#t].

Reestructurar este diagrama en base a agregaciones, teniendo en cuenta las dependencias de ORG para garantizar equivalencia semántica, podría resultar en un esquema como el indicado en la figura siguiente

---

---

---

---

---

---

---

---

---

---





### Ventajas y desventajas de las transformaciones.

- Las transformaciones que resultaron factibles tienen como base o permiten encontrar un esquema ER convencional o extendido con agregaciones, solamente. No ha sido necesario introducir nueva notación.
- Los casos restantes no pueden ser transformados de manera tal que la situación original sea mejorada en algún aspecto (operativo, nivel semántico, simplicidad).
- Dado que los esquemas transformados generalmente introducen nuevas tablas, este aspecto debería ser cuidadosamente analizado teniendo en cuenta las características del UdeD (estabilidad de la información, frecuencia de actualización, etc.)

---

---

---

---

---

---

---

---

### Dependencias de Exclusión.

Una dependencia de exclusión (*de*) es una restricción que expresa que dos relaciones, o dos proyecciones sobre atributos compatibles de las mismas, son disjuntas. Una *DE* puede expresarse en SQL:1999 mediante aserciones o triggers. Una forma alternativa, es mediante sendos chequeos de tabla:

- en  $R_i$ : CHECK (( $R_i.W_i$ ) NOT IN (Select  $W_j$  FROM  $R_j$ ))
- y en  $R_j$ : CHECK(( $R_j.W_j$ ) NOT IN (Select  $W_i$  FROM  $R_i$ ))

para evitar los problemas de inconsistencia que potencialmente permiten los chequeos con subconsultas asociados a una tabla.

---

---

---

---

---

---

---

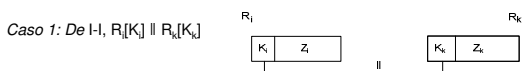
---

### Dependencias de Exclusión.

Así como una partición vertical de una tabla genera tácitamente una dependencia de igualdad, una partición horizontal lo hace respecto de una dependencia de exclusión.

Este concepto no se representa en un *DER* convencional como tal. Sin embargo puede especificarse de manera similar a una jerarquía de especialización (exclusiva).

En estas abstracciones, la pertenencia a una subclase induce tácitamente una dependencia de exclusión respecto de las otras subclases.



---

---

---

---

---

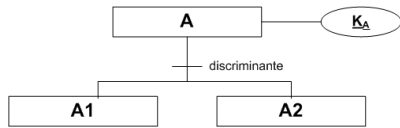
---

---

---

### Dependencias de Exclusión.

- Una abstracción del MER convencional donde aparecen las dependencias de exclusión → jerarquías de especialización/generalización exclusivas (totales o parciales)



- $A1[K_A] \parallel A2[K_A]$

---

---

---

---

---

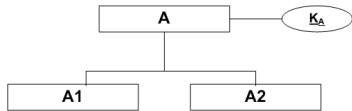
---

---

---

### Dependencias Superposición Parcial.

- Infrecuentes como especificaciones de reglas del negocio.
- Un caso en el que pueden encontrarse fáctitas → jerarquías de especialización/generalización compartidas (totales o parciales)



- $A1[K_A] \cap A2[K_A] \rightarrow$  estrictamente compartida (la semántica de la jerarquía no lo exige)

---

---

---

---

---

---

---

---