

Especificación de Restricciones de Integridad Condicionales en Esquemas Conceptuales de Bases de Datos

Jorge Carrasco, Marcela Varas
{jcarrasc, mvaras}@inf.udec.cl

Departamento de Ingeniería Informática y Ciencias de la Computación
Universidad de Concepción
Chile.

Resumen

En los esquemas conceptuales de bases de datos, se debe especificar lo más completamente posible el dominio de la aplicación. Esto es aún más complejo cuando esta realidad presenta excepciones, las cuales no se pueden especificar adecuadamente con los lenguajes de modelación tradicionales, como el modelo entidad interrelación.

En este artículo se presenta una propuesta para la especificación de restricciones de integridad condicionales, las cuales son adecuadas para especificar situaciones que se derivan de la ocurrencia de excepciones, en las cuales una determinada restricción de integridad deja de ser válida. Para ello, se proponen los elementos básicos que constituirán el lenguaje y se ilustra su aplicación mediante el uso de ejemplos.

Palabras Clave

Diseño Conceptual de Bases de Datos, MER Extendido, Restricción de integridad, Restricción de integridad condicional.

1. Introducción

Cuando se modela una realidad (área de negocio) en el nivel conceptual, junto con identificar los datos que se desea manejar y cómo éstos se relacionan, surgen 'reglas' que deben cumplir dichos datos para representar lo más fielmente posible esa realidad. Es decir, el cumplimiento de estas reglas ayuda a mantener la integridad de los datos. Estas reglas se reflejan en la especificación de la base de datos (esquema) a través de las Restricciones de Integridad de distintas maneras, como por ejemplo: definición de dominios para los datos, especificación de cardinalidad en los tipos de interrelación, etc.

Existen algunas restricciones propias de los datos que son difíciles, si no imposibles de representar utilizando los modelos tradicionales. Esto produce una pérdida de expresividad en los esquemas y una falta de portabilidad de los mismos, ya que en las etapas posteriores, aquellas restricciones son manejadas por las aplicaciones software que usan dichos datos, o por el administrador de bases de datos, mediante procedimientos almacenados y/o gatilladores (triggers) los cuales deben ser programados por los desarrolladores y quedan documentados separadamente del modelo conceptual y son considerados en etapas tardías del desarrollo.

La importancia de considerar las restricciones que afectan a los datos que se están modelando, radica en que la satisfacción de éstas conforma una garantía de la consistencia de la base de datos en el tiempo. Ahora bien, mientras antes se consideren las restricciones en el desarrollo, antes se podrán detectar errores, depurar y modificar los esquemas de la base de datos para asegurar su cumplimiento. La consideración de las restricciones en etapas posteriores puede inducir a modificaciones costosas del diseño [Varas96].

Este hecho justifica todos los esfuerzos por incluir formalmente las restricciones en las etapas iniciales del desarrollo, ya que de otra manera quedarán, en el mejor de los casos, expresadas sólo en lenguaje natural, con el consiguiente riesgo de que no sean consideradas en la implementación final.

Se propone en el presente trabajo un lenguaje de representación para las restricciones condicionales, cuya aplicación o forma de aplicación esta supeditada, como su nombre lo indica, al cumplimiento de alguna condición. Estas condiciones pueden estar determinadas por el valor de algún elemento del esquema (por ejemplo el valor de un atributo), en cuyo caso se les llama "condiciones dependiente del contenido", o bien de una situación externa al esquema (por ejemplo la fase en que se encuentra la base de datos), en cuyo caso se les llama "condiciones independiente del contenido"

2. Restricciones de Integridad

Las Restricciones de Integridad (RI) estáticas se definen como aquellas que condicionan los estados válidos que puede tomar la BD, es decir, los valores que le son permitidos; las RI dinámicas, son aquellas que condicionan las transiciones válidas en la BD, es decir, dado el valor actual de la BD, cuáles son los posibles valores que puede tomar a continuación. Las RI estáticas que se incluyen en la mayoría de los modelos ER ([Batini94, deMiguel93]) son:

1. Dominio
2. Cardinalidad de un atributo con respecto a tipo de entidad.
3. Cardinalidad de un tipo de entidad con respecto a un tipo de interrelación
4. Identificadores
5. Generalización
6. Cobertura
7. Tipos de Interrelación exclusivas con respecto a un tipo de entidad
8. Dependencia entre atributos

De este conjunto de restricciones (estáticas), nos abocaremos al análisis y especificación de restricciones condicionales que involucren a las restricciones de cardinalidad.

3. Hacia un lenguaje de especificación.

Se definen a continuación los elementos básicos del lenguaje propuesto.

3.1 Valor

Esta es una función que tiene por objeto obtener el valor de un atributo de un tipo de entidad o un tipo de interrelación. Si T es un tipo de entidad o un tipo de interrelación que tiene a a_1, \dots, a_n como atributos.

$$\text{Valor}(T, a_i) = \text{valor}$$

obtiene el valor de a_i , con $i \in \{1, \dots, n\}$, donde $\text{valor} \in D$, con D, dominio de a_i .

3.2 Fase

Propiedad de los objetos de la base de datos que condiciona la aplicación de las restricciones de integridad. Las fases a considerar serán la fase normal y la fase de excepción. De este modo, para un tipo de entidad o tipo de interrelación T:

$$\text{Fase}(T) = \text{fase}$$

donde fase \in {normal, excepción}. *Normal* indica que la BD se encuentra a régimen y, por lo tanto, las RI se aplican en forma normal y, *excepción*, indica que debido al estado de la BD, las RI pueden no ser aplicables. Ejemplos de fase de excepción son cuando la base de datos está siendo poblada y cuando se está en proceso de recuperación de información histórica. También se puede utilizar la fase de excepción para usuarios con amplios privilegios.

3.3 Proyección*

El efecto producido por el operador proyección es la definición de una nueva estructura, eliminando un conjunto de atributos (o componentes de un atributo complejo) desde un TE (TI). Si un tipo de entidad o un tipo de interrelación T tiene a $a_1, a_2, \dots, a_i, \dots, a_k, \dots, a_n$ como atributos,

$$T' = \Pi T (a_i, a_{i+1}, \dots, a_k), i \geq 1, n \geq k.$$

T' es definido como el nuevo tipo de entidad o tipo de interrelación (dependiendo si T es tipo de entidad o tipo de interrelación respectivamente), que tiene k-i+1 atributos a_i, a_{i+1}, \dots, a_k , siendo los mismos atributos de E_1 , pero omitidos los atributos (a_1, \dots, a_{i-1}) y (a_{k+1}, \dots, a_n) .

3.4 Restricción*

La restricción permite obtener instancias de un tipo de entidad o un tipo de interrelación T frente a un predicado de selección c de la forma:

$$\langle \text{operando} \rangle \langle \text{operador de comparación} \rangle \langle \text{operando} \rangle$$

donde, operando es un nombre de atributo, un valor o un conjunto de valores y el operador de comparación es un operador relacional ($=, \neq, <, \leq, >, \geq$). Además, c puede ser el conjunto de predicados de selección unidos por los operadores booleanos (\neg, \wedge, \vee). Se denota como:

$$T' = \sigma T (c)$$

definiendo un nuevo tipo de entidad o tipo de interrelación T' (dependiendo si T es tipo de entidad o tipo de interrelación, respectivamente), cuyas instancias son aquellas que pertenecen a T y que satisfacen la condición c .

3.5 Contar

Esta función permite contar el número de instancias de un tipo de entidad o tipo de interrelación T.

$$\text{Contar}(T) = \text{total}$$

con total entero no negativo.

3.6 Estructura de Selección

Basándonos en la definición de instrucción condicional de los lenguajes de programación [Pratt96], tenemos:

```

SI  $c_1$  ENTONCES
  <Restricción 1>
SI  $c_2$  ENTONCES
  <Restricción 2>
...
SI NO

```

* Esta definición está basada en la definición presentada en [Guti99]

<Restricción n>
FIN SI

donde los c_i son predicados (como los definidos para la operación de restricción definida anteriormente) y $\langle \text{Restricción}_i \rangle$ con $i \in \{1, n-1\}$, es la definición de la restricción para el caso que se cumpla c_i . En caso de que no se cumpla ninguna condición c_i , se aplica restricción n.

3.7 Comentario.

Las funciones Valor, Contar, Proyección*, Restricción*, se utilizan para definir restricciones condicionales con condiciones dependientes del contenido (ver ejemplos apartado 4.).

La función Fase, se utiliza para especificar restricciones condicionales con condiciones independientes del contenido.

La estructura de selección se utiliza para la definición de toda restricción condicional.

4. Ejemplos

A continuación se presentan algunos casos para ilustrar la ocurrencia de excepciones y la aplicación de restricciones condicionales para su manejo.

4.1 Caso Institución Previsional

En una institución de salud previsional, todos los beneficiarios de un afiliado deben tener RUT. Sin embargo, debe considerarse un caso excepcional: un hijo de un afiliado pasa a ser su beneficiario desde el séptimo mes de embarazo, momento en el que, por supuesto, no tiene RUT, ya que todavía no está inscrito en el registro civil.

Primera representación:

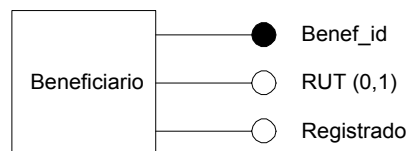


Figura 1

donde el atributo 'registrado' indica si está o no inscrito en el Registro Civil, cuyo dominio sería ('si', 'no').

Este ejemplo considera una restricción de cardinalidad de un atributo sobre un TE (TI) dependiendo de otro atributo del mismo TE (TI). En este caso:

$$\text{Card}(\text{Rut}, \text{Beneficiario})=(0,1)$$

Sin embargo, de esta manera no queda especificado que sólo los beneficiarios no registrados pueden carecer de RUT.

Segunda representación:

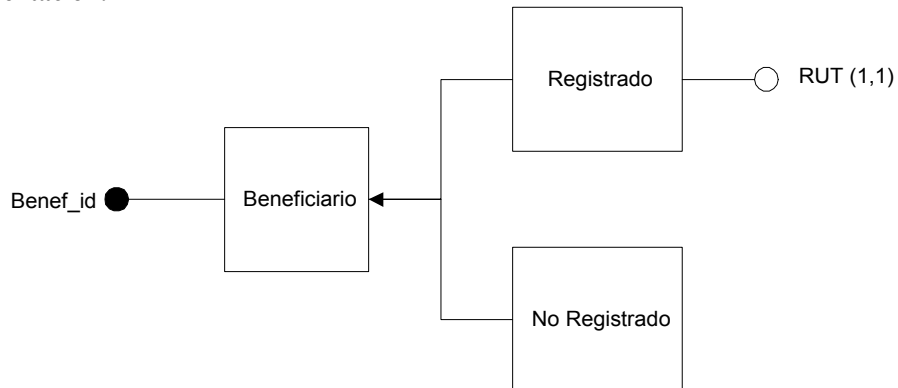


Figura 2

Representación con restricción condicional:

Basándonos en la figura 1 y usando la función Valor definida anteriormente, podemos expresar la restricción en forma condicional como sigue:

```
SI Valor(Beneficiario, registrado)='si' ENTONCES  
    Card(rut,Beneficiario)=(1,1)  
SI NO  
    Card(rut,Beneficiario)=(0,0)  
FIN SI
```

Discusión:

En este caso se puede observar que una representación utilizando estructuras de generalización (también denominada de subtipos) es una alternativa para una restricción condicional.

En situaciones como estas, con condiciones dependientes del contenido, la incorporación de restricciones condicionales simplifica la estructura del esquema.

4.2 Caso Video Club.

En un video club, los clientes están categorizados en normales, buenos, o excelentes y, de acuerdo a esto, en cada arriendo que realicen, pueden llevar, a lo más, 1, 2 o 3 películas respectivamente.

Primera representación:



Figura 3

En este ejemplo se impone una regla que norma la cantidad de veces que puede participar un TE en un TI dependiendo de un atributo del TE.

$Card(\text{Cliente}, \text{Arrienda})=(0,3)$

Sin embargo, esta restricción no considera que la cardinalidad máxima es dependiente de la categoría del cliente.

Segunda representación:

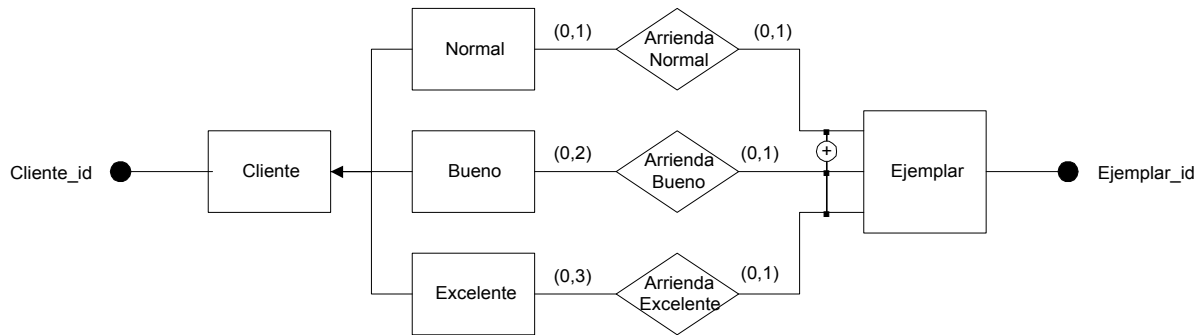


Figura 4

En este caso se muestra correctamente la cantidad máxima de videos que puede arrendar un cliente según su tipo.

Representación con restricción condicional:

Redefiniendo la restricción ocupada en la figura 3 en forma condicional tenemos:

SI Valor(Cliente, categoría)=normal ENTONCES
 Card(Cliente,Arriendo)=(0,1)
 SI Valor(Cliente, categoría)=bueno ENTONCES
 Card(Cliente,Arriendo)=(0,2)
 SI Valor(Cliente, categoría)=excelente ENTONCES
 Card(Cliente,Arriendo)=(0,3)
 FIN SI

Discusión:

Al igual que en el caso expuesto en el apartado 4.2, la incorporación de una restricción condicional simplifica el esquema (en su parte estructural). Esto no sería conveniente si las tres categorías de Cliente (Normal, Bueno, Excelente) tuviesen más diferencias entre si, como por ejemplo, otros atributos o otros tipos de interrelación involucrados.

4.3 Caso Estructura Organizacional

Una empresa cuenta con un cierto número de empleados. Todos los empleados tienen asignado uno y sólo un jefe directo, con excepción del Gerente General

Primera representación:

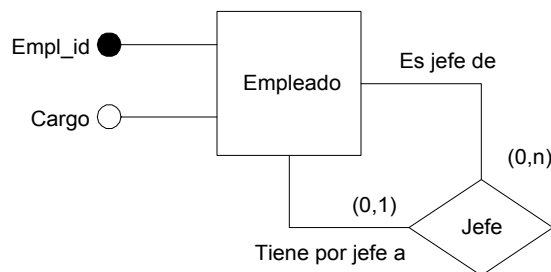


Figura 5

donde

$\text{Card}(\text{Es jefe de, Jefe})=(0,n)$
 $\text{Card}(\text{Tiene por jefe a, Jefe})=(0,1)$

En este caso no se especifica que sólo el Gerente General no tiene un jefe directo.

Segunda representación:

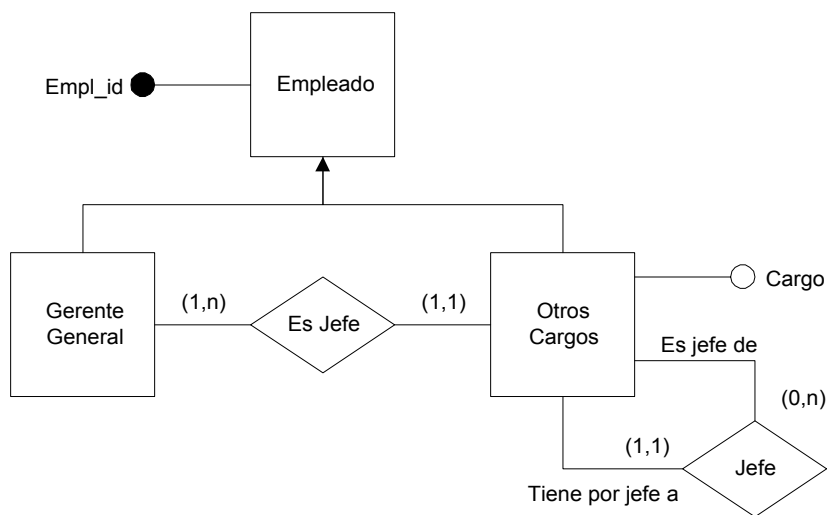


Figura 6

Si bien esta representación es mejor que la anterior ya que sólo el Gerente General está sin jefe directo, no asegura que un empleado con otro cargo esté tanto en el TI 'Es jefe' y en 'Jefe' teniendo, por consiguiente, dos jefes directos, situación que no puede darse. Una mejor solución sería cambiar $\text{Card}(\text{Otros cargos, Es jefe})=(0,1)$ y $\text{Card}(\text{Tiene por jefe a, Jefe})=(0,1)$ y dejándolas como tipos de interrelaciones exclusivas, sin embargo esto permitiría que algún empleado con otro cargo quedara sin jefe directo.

Representación con restricción condicional:

Ocupando la representación mostrada en la figura 5, podemos definir la restricción como sigue:

```
SI Valor(Empleado, cargo)='Gerente General' ENTONCES
    Card(Tiene por jefe a, Jefe)=(0,0)
SI NO
    Card(Tiene por jefe a, Jefe)=(1,1)
FIN SI
```

Discusión:

En este ejemplo se aprecia una situación que es muy compleja de modelar utilizando el modelo entidad relación, y que se simplifica notablemente al introducir una restricción condicional.

4.4 Caso Reasignación de Personal.

En una empresa cualquiera, se tienen empleados asignados a Departamentos.

Primera representación:



Figura 7

En este caso se considera que un empleado siempre debe estar asignado a un Departamento. Además, se explicita que todo departamento debe tener asignado empleados (al menos uno).

Estas reglas no son válidas en al menos las siguientes situaciones:

1. Un empleado está en una situación transitoria en la cual no está asignado a ningún departamento.
2. Hay empleados que no están asignados a ningún departamento, pero representan un proporción muy baja del total de empleados.

Segunda representación:

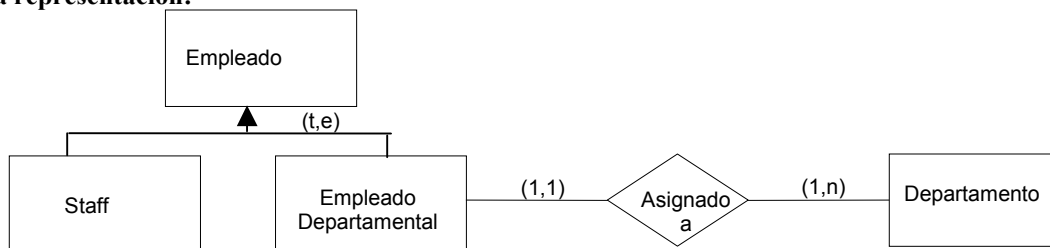


Figura 8

En este caso se ha resuelto el caso de los empleados de staff, pero no se ha resuelto el problema de los Empleados transitoriamente sin asignación.

Representación con restricción condicional:

Redefiniendo la restricción ocupada en la figura 8 en forma condicional tenemos:

```
SI fase(Empleado Departamental)=normal ENTONCES  
    Card(Empleado Departamental, Asignado a)=(1,1)  
SINO  
    Card(Empleado Departamental, Asignado a)=(0,1)  
FIN SI
```

Discusión:

En este ejemplo se ilustra la incorporación de una restricción condicional con condición independiente del contenido. De este modo las dos excepciones a la cardinalidad de Empleado con respecto a Asignado a quedan completamente especificadas utilizando el lenguaje propuesto.

La excepción que fue representada por una estructura de generalización también pudo ser expresada mediante una restricción condicional, pero ese caso ya fue ilustrado previamente.

5. Conclusiones y Comentarios.

Se presenta un lenguaje de especificación para restricciones condicionales, que se constituye en una alternativa válida para representar ciertas situaciones para las cuales el MER extendido tiene una representación (que puede ser natural o forzada) dando, por consiguiente, mayor libertad al diseñador. Existen casos en que la no posibilidad de expresar una excepción (mediante una restricción condicional) simplemente deja incompleta la especificación del esquema (ver ejemplo de la sección 4.3).

En términos genéricos, la incorporación de restricciones condicionales con condiciones dependientes del contenido, simplificará el esquema conceptual. Esto último en cuanto a su estructura. Este tipo de restricciones condicionales pueden incorporarse en la estructura de un esquema utilizando estructuras de generalización.

Las restricciones condicionales con condiciones independientes del contenido no pueden representarse en la estructura del esquema utilizando el modelo entidad relación. Aquí el uso de estructuras de generalización no es capaz de expresar este tipo de situaciones.

La posibilidad de expresión de restricciones condicionales permite la especificación más completa de bases de datos, otorgándole flexibilidad a los esquemas (a través de una flexibilización de sus restricciones). Esto es de gran importancia, pues esta flexibilización da cuenta de la tolerancia que el sistema tiene a las excepciones, las cuales están presentes en toda realidad.

Se provee de constructores usables, con similitudes al álgebra relacional y a los lenguajes de programación, para facilitar su aprehensión por parte de los diseñadores. Las restricciones condicionales no son más que abreviaciones para fórmulas lógicas. El aporte de presentarlas a través de constructores sencillos es que posibilita su uso en etapas tempranas del desarrollo por diseñadores y usuarios que no poseen soltura en notaciones formales, y para los cuales las fórmulas lógicas pueden resultar crípticas. En la práctica, los diseños de los sistemas se especifican utilizando sólo los lenguajes que proveen las herramientas de modelación, y esto no incluye necesariamente a la lógica.

El lenguaje propuesto es para el nivel conceptual. Las especificaciones generadas deberán mapearse a especificaciones implementables en alguna plataforma de implementación (como algún SGBD). La definición de estas transformaciones forma parte de las proyecciones de este trabajo, junto con una mayor profundización en el tema para asegurarse que el lenguaje propuesto es capaz de considerar otras restricciones de integridad, como aquellas consideradas por el modelo CCER [Varas98] y no sólo las de cardinalidad. Es necesario además, identificar otras situaciones que puedan definir una condición para la aplicación de una restricción con el fin de formalizarla y poder incluirla en el lenguaje.

6. Bibliografía y Referencias

- [Batini94] C. Batini, S. Ceri, S. Navathe, "Diseño Conceptual de Bases de Datos: Un enfoque de entidades-interrelaciones", Addison-Wesley/Díaz de Santos, 1994.
- [deMiguel93] A. de Miguel, M. Piattini, "Concepción y Diseño de Bases de Datos: del Modelo E/R al Modelo Relacional", Addison-Wesley Iberoamericana, 1993.
- [Guti99] Emerson Gutierrez, "Operaciones para el Modelo CCER", Memoria de Título para optar al Título de Ingeniero Civil Informático, Universidad de Concepción, 1999
- [Pratt96] Terrence Pratt, Marvin Zelkowitz, "Programming Languages, Design and Implementation", Prentice-Hall, 3era edición, 1996

- [Varas96] Marcela Varas, Christian Weldt, Claudia Jimenez, "Una Propuesta para la Especificación de Restricciones Dinámicas en Esquemas Conceptuales de Bases de Datos", IV Encuentro Chileno de Computación, 1996
- [Varas98] Marcela Varas, "Formalismo de representación de restricciones de integridad en esquemas conceptuales de bases de datos", Tesis para optar al grado de magister en ciencias de la computación, Universidad de Concepción, 1998