Proyecto de Normalización Automática de Base de Datos

Lic. Beatriz Steimberg *

Resumen

En el primer cuatrimestre del año 2003 se encaró el proyecto de Normalización Automática de Base de Datos.

El objetivo de este proyecto de investigación fue generar una pieza de software que permitiese realizar en forma automática el pasaje de un modelo conceptual de base de datos a un modelo lógico en el que las relaciones cumpliesen con estándares de calidad que, bajo el paraguas teórico de Formas Normales, evitasen redundancias y anomalías de actualización.

1. ¿Qué es una base de datos?

Si analizamos los circuitos de la información en una gran empresa, con seguridad encontraremos que los datos que surgen de sus procesos operativos se encuentran volcados en una o varias bases de datos. La conclusión será la misma si nos dirigimos a una empresa pequeña , a un organismo de la administración pública en cualquiera de sus niveles, a una biblioteca de una universidad o a la AFIP.

¿Qué es entonces una base de datos?

En su concepción vulgar no es otra cosa que un conjunto de datos, al estilo de una agenda con apellidos y teléfonos, o un catálogo de precios o los legajos del personal de una empresa.

En la acepción que nosotros le daremos, las bases de datos surgen a fines de los años 60, como respuesta a la anarquía que planteaba en las organizaciones la existencia de una cada vez mayor cantidad de archivos, cada vez más extensos, con información redundante entre ellos. La propuesta que trae la tecnología de base de datos es la siguiente:

- Tomar el conjunto de datos que son relevantes para toda la organización
- Organizarlos correctamente
- Colocarlos en un reservorio único: la base de datos
- Impedir que los programas accedan directamente a registros o campos; entre ellos y los datos reales colocar una pieza de software compleja, un sistema de gestión de base de datos, de manera de
    - o aislar a los programas de cambios que pudieran producirse en la estructura de la base de datos

* Docente de la Facultad de Ingeniería. Universidad de Palermo.

- garantizar una serie de funciones adicionales que sólo mencionaremos a título informativo: integridad de datos, independencia física y lógica, manejo de transacciones, recupero ante fallas, seguridad,....

Una base de datos entonces es un conjunto de datos persistentes utilizados por los sistemas de aplicaciones de una empresa determinada

Qué hace falta para poder "explotar" una base de datos? En primer lugar diseñarla.

2. Etapas en el diseño de una base de datos

El punto de partida es representar en forma abstracta y simplificada la porción de la realidad que nos interesa: construir el modelo que luego manipularemos. Es el momento en que se definen entidades –representación de un objeto del mundo real creado usando los valores de sus propiedades significativas en forma computable- y sus interrelaciones, que constituyen el **modelo conceptual.**

Las propiedades significativas de las entidades son los atributos y tienen dos características para nosotros fundamentales:

- algunos colaboran en distinguir a la entidad a la que se aplican de otras de la misma clase: son atributos clave.
- Entre algunos de ellos existen relaciones de dependencia funcional, siempre semánticas o propias del significado que las cosas tienen en ese modelo, que constituyen restricciones sobre las tuplas que pueden aparecer en una relación:

  Se dice que un atributo X depende funcionalmente de otro Y,
  $$X \rightarrow Y,$$
  si y solo si a cada valor del atributo X le corresponde un único valor de Y.

Cumplida la etapa anterior, es el momento del traslado del modelo conceptual a un **modelo lógico**, que pueda ser implementado en un computador. En el mercado actual el enfoque dominante es el relacional, que tiene una sólida base matemática . Una base de datos relacional consiste en un conjunto de tablas o relaciones, con filas o registros y columnas o atributos.

A modo de ejemplo, está podría ser la relación o tabla Alumnos.

Legajo	*Nombre*	*Apellido*
1	Marcos	Perez
2	Lucas	Lopez
3	Marta	Gozalez

Y el esquema que le corresponde a la misma tabla es el siguiente:

Alumnos (Legajo, Nombre, Apellido)

Un tema central en el diseño lógico de la base de datos es cómo estructurar las tablas que la constituyen de la mejor forma posible, logrando menor cantidad de datos, menor tamaño de base de datos y actualizaciones en un solo lugar. En síntesis, cómo testear el resultado de un esquema de base de datos construido intuitivamente?

La **normalización es justamente el proceso estandarizado de reducción de un conjunto de relaciones a formas más deseables**, evitando:

- La redundancia de los datos: repetición de datos en un sistema.
- Anomalías de actualización: inconsistencias de los datos como resultado de datos redundantes y actualizaciones parciales.
- Anomalías de borrado: pérdidas no intencionadas de datos debido a que se han borrado otros datos.
- Anomalías de inserción: imposibilidad de adicionar datos en la base de datos debido a la ausencia de otros datos.

A modo de ejemplo veamos los problemas que se plantean sobre la tabla Libro (Autor, Nacionalidad, Codigo_libro, Titulo_libro, Editor)

- Redundancia: cuando un autor tiene varios libros, se repite innecesariamente su nacionalidad.
- Anomalías de inserción: no se puede dar de alta un autor sin libros.

La normalización supone un espacio de exigencias crecientes, bajo la modalidad de distintas formas normales (primera, segunda, ...), de forma tal que una relación estará en la forma normal n+1 sólo si lo está en la forma n y satisface requisitos adicionales. Y una base de datos estará en la forma n sólo si todas las tablas que la constituyen se encuentran en esa forma normal.

3. Objetivo del Proyecto de Investigación

En el primer cuatrimestre del año 2003 se encaró el proyecto de Normalización Automática de Base de Datos.

El objetivo de este proyecto de investigación fue generar una pieza de software que permitiese **realizar en forma automática el pasaje de un modelo conceptual de base de datos a un modelo lógico.** O sea, contando con los objetos significativos para el modelo y sus correspondientes atributos, más el conjunto de dependencias funcionales, **generar una base de datos relacional normalizada.**

Existen en el mercado una serie de productos que asisten al informático y a los expertos en dominio en la construcción del modelo conceptual. Otros lo hacen en la elaboración de la interface entre el modelo conceptual y el lógico, derivando, a partir de una representación gráfica del primero, las dependencias funcionales o vinculaciones lógicas entre atributos.

Nuestro enfoque apuntó al diseño lógico propiamente dicho, convencidos de que la implementación de software de este tipo posibilitaría a las organizaciones que lo utilicen reducir considerablemente el tiempo que habitualmente destinan al modelado de sus bases de datos y les garantizaría contar con bases normalizadas, de forma de evitar los inconvenientes derivados de consulta y actualización a bases no normalizadas

El software que desarrollamos cubre:

- La transformación del conjunto de dependencias funcionales propuestas por el usuario en un conjunto equivalente pero sin redundancias [Fm]
- La determinación de la/s clave/s de la relación inicial
- La normalización del esquema inicial hasta la 3FN mejorada (BCNF), previéndose la extensión del producto hasta la quinta forma normal (5FN)

Se ha buscado optimizar los algoritmos utilizados, que responden a procesos complejos, con fundamentos en mecanismos de inteligencia artificial y con alto grado de recursividad, para evitar un excesivo consumo de recursos (espacio en disco y tiempo de procesamiento).

Se utilizó como lenguaje de programación Borland C++ Versión 3.3.

ANEXO

Muy brevemente, desarrollamos a continuación el alcance de las primeras tres formas normales, por ser las más usadas y las de comprensión más intuitiva.

Primera Forma Normal (1NF)

Una relación se encuentra en primera forma normal (1NF) si y solo si cada uno de sus atributos contiene un único valor para un registro determinado.

Supongamos que deseamos guardar los cursos que están realizando los alumnos de un determinado centro de estudios; podríamos considerar el siguiente diseño:

Legajo	*Nombre*	*Apellido*	*Cursos*
1	Marcos	Perez	Inglés
2	Lucas	Lopez	Contabilidad, Informática
3	Marta	Gozalez	Inglés, Contabilidad

Podemos observar que el registro de Legajo = 1 cumple la primera forma normal, pero no ocurre así con los registros identificados por los legajos 2 y 3, ya que en ambos casos el campo Cursos contiene más de un dato. La solución en este caso es crear dos tablas del siguiente modo:

Tabla A		
Legajo	***Nombre***	***Apellido***
1	Marcos	Perez
2	Lucas	Lopez
3	Marta	Gozalez

Tabla B	
Código	***Curso***
1	Inglés
2	Contabilidad
2	Informática
3	Inglés
3	Informática

Como se puede comprobar, en el nuevo esquema los registros de ambas tablas contienen valores únicos en sus campos, por lo tanto ambas tablas cumplen la primera forma normal o, lo que es lo mismo, el esquema actual está en 1FN

Segunda Forma Normal (2FN)

Una relación está se encuentra en segunda forma normal (2FN) si y solo si cumple 1FN y todos sus atributos no clave dependen en forma completa de la clave.

Supongamos que construimos una tabla con los años que cada empleado ha estado trabajando en cada departamento de una empresa:

Codigo_Empleado	***Codigo_Dpto.***	***Apellido_Nombre***	***Departamento***	***Anios***
1	6	Juan García	Contabilidad	6
2	3	Pedro Paglione	Sistemas	3
3	2	Sonia Ballesteros	I+D	1
4	3	Verónica Paniza	Sistemas	10
2	6	Pedro Paglione	Contabilidad	5

La clave de esta tabla está formada por los campos Codigo_Empleado y Codigo_Departamento y la relación se encuentra en 1FN.

1. El campo Apellido_Nombre no depende funcionalmente de toda la clave, sólo depende de Codigo_Empleado.
   Codigo_Empleado → Apellido_Nombre
2. El campo Departamento no depende funcionalmente de toda la clave, sólo del campo Codigo_Departamento.
   Codigo_Departamento → Departamento
3. El campo Anios (representa el número de años que cada empleado ha trabajado en cada departamento) depende funcionalmente de la clave en forma completa Codigo_Empleado, Codigo_Departamento → Anios

Por lo expresado en 1. y 2., no se cumple 2FN. La solución es la siguiente:

Tabla A

Código Empleado	*Apellido_Nombre*
1	Juan García
2	Pedro Paglione
3	Sonia Ballesteros
4	Verónica Paniza

Tabla B

Codigo _Departamento	*Departamento*
2	I+D
3	Sistemas
6	Contabilidad

Tabla C

Codigo_Empleado	*Codigo_Departamento*	*Anios*
1	6	6
2	3	3
3	2	1
4	3	10
2	6	5

Podemos observar que ahora las tres tablas, cuyas claves son respectivamente Codigo_Empleado, Codigo_Departamento y los campos Codigo_Empleado y Codigo_Departamento, se encuentran en segunda forma normal.

Tercera Forma Normal (3FN)

Una relación se encuentra en 3FN si y solo si está en 2FN y los campos no clave dependen únicamente de la clave o, dicho en otras palabras, los campos no clave no dependen unos de otros.

Tomando como referencia el ejemplo anterior, y suponiendo que cada alumno sólo puede realizar un único curso a la vez y que deseamos guardar información sobre el aula en que se imparte el curso. Podemos plantear la siguiente estructura:

Legajo	*Apellido_Nombre*	*Curso*	*Aula*
1	Juan García	Informática	Aula A
2	Pedro Paglione	Inglés	Aula B
3	Sonia Ballesteros	Contabilidad	Aula C

Estudiando las dependencias de cada campo con respecto a la clave Legajo surgen las siguientes dependencias funcionales:

- Legajo → Apellido_Nombre.
- Legajo → Curso.
- Legajo → Aula
- Pero Aula, que depende funcionalmente de Legajo, está también ligada al curso que el alumno está realizando. O sea se cumple la siguiente dependencia funcional: Curso → Aula

Por esta última razón se dice que la tabla no está en 3FN. La solución es la siguiente:

Tabla A		
Código	*Apellido_Nombre*	*Curso*
1	Juan García	Informática
2	Pedro Paglione	Inglés
3	Sonia Ballesteros	Contabilidad

Tabla B	
Curso	*Aula*
Informática	Aula A
Inglés	Aula B
Contabilidad	Aula C