Unidad IV: Diseño de bases de datos relacionales

4.1 Características del diseño relacional

Es un nivel de abstracción más bajo que el modelo E-R y es la representación en tablas (esquema) del problema, el cual es práctimente un paso antes del nivel físico.

En la unidad anterior se mencionaron 3 tipos de modelado: conceptual, lógico y físico.

El modelo E-R se considera un modelo conceptual ya que permite a un nivel alto el ver con claridad la información utilizada en algun problema o negocio.

En esta unidad nos concentraremos en desarrollar un buen modelo "lógico" que se conoce como "esquema de la base de datos" (*database schema*) a partir del cual se podrá realizar el modelado físico en el DBMS, es importante mencionar que es un paso necesario, no se puede partir de un modelo conceptual para realizar un físico.

Puede resultar confuso el concepto de modelo Entidad-Relación vs Modelo Relacional, quizás porque ambos comparten casi las mismas palabras. Como se mencionó en la anteriormente, el objetivo del **Modelo Relacional** es crear un "esquema" (*schema*), lo cual como se mencionará posteriormente consiste de un conjunto de "tablas" que representan "relaciones", relaciones entre los datos.

Estas tablas, pueden ser construídas de diversas maneras:

- Creando un conjunto de tablas iniciales y aplicar operaciones de normalización hasta conseguir el esquema más óptimo. Las técnicas de nomalización se explican más adelante.
- Convertir el diagrama E-R a tablas y posteriormente aplicar también operaciones de normalización hasta conseguir el esquema óptimo.

La primer técnica fue de las primeras en existir y, como es de suponerse, la segunda al ser más reciente es mucho más conveniente en varios aspectos:

- El partir de un diagrama visual es muy útil para apreciar los detalles, de ahí que se llame modelo conceptual.

- El crear las tablas iniciales es mucho más simple a través de las reglas de conversión.
- Se podría pensar que es lo mismo porque finalmente hay que "normalizar" las tablas de todas formas, pero la ventaja de partir del modelo E-R es que la "normalización" es mínima por lo general.
- Lo anterior tiene otra ventaja, aún cuando se normalice de manera deficiente, se garantiza un esquema aceptable, en la primer técnica no es así.

4.2 Dominios atómicos y la primera forma normal

Cuando pasamos al modelo relacional debemos aplicar ciertas reglas las de estandarización, de normalización del todo las tablas, a este conjunto de reglas se le conoce con el nombre de normalización de base de datos, que consiste en aplicar una serie de relación las relaciones obtenidas tras el paso del modelo entidad relación al modelo relacional. Las base de datos relacionales se normalicen para evitar redundancia de los datos, evitar problemas de actualización de los datos en las tablas, para proteger la integridad de los datos. Aquí es importante mencionar que tenemos tres tipo de integridades, la integridad de dominio que como dijimos anteriormente limita el conjunto de datos posibles en una columna. La integridad de identidad, que establece que cada fila debe ser única y que no se permiten la duplicidad. La integridad referencial que plantean que cuando un atributo columna de una tabla hace referencia a la información de la tabla.

4.3 Dependencias funcionales

Las dependencias funcionales son restricciones del conjunto de relaciones legales. Permiten expresar hechos
sobre la empresa que se modela con la base de datos.

En el Capítulo 2 se definió el concepto de superclave de la manera siguiente. Sea

R el esquema de una relación.
El subconjunto K de R es una superclave de R si, en cualquier relación legal r(R), para todos los pares t1
y t2 de tuplas de r tales que t1 ≠ t2, t1 [K] ≠ t2 [K]. Es decir, ningún par de tuplas de una relación legal r(R)
puede tener el mismo valor para el conjunto de atributos K.

El concepto de dependencia funcional generaliza la noción de superclave. Considérese el esquema de una
relación R y sean α ⊆ R y β ⊆ R. La dependencia funcional

α →β

se cumple para el esquema R si, en cualquier relación legal r (R), para todos los pares de tuplas t1 y t2 de r tales
que t1 [α] = t2 [α], también ocurre que t1 [β] = t2 [β].

Empleando la notación para la dependencia funcional, se dice que K es una superclave de R si K →R. Es
decir, K es una superclave si, siempre que t1 [K] = t2 [K], también se produce que t1 [R] = t2 [R] (es decir, t1 = t2).

Las dependencias funcionales nos permiten expresar las restricciones que no se pueden expresar con las
superclaves.

4.4 Segunda forma normal

La segunda forma normal (2NF) es una forma normal usada en normalización de bases de datos. Una tabla que está en la primera forma normal (1NF) debe satisfacer criterios adicionales para calificar para la segunda forma normal.

Una tabla 1NF estará en 2NF si y solo si, dada una clave primaria y cualquier atributo que no sea un constituyente de la clave primaria, el atributo no clave depende de **toda** la clave primaria en vez de solo una parte de ella.

4.5 Tercera forma normal

La tercera forma normal (3NF) es una forma normal usada en la normalización de bases de datos. La 3NF fue definida originalmente por E.F. Codd[1] en 1971. La definición de Codd indica que una tabla está en 3NF si y solo si las dos condiciones siguientes se cumplen:

- La tabla está en la segunda forma normal (2NF)
- Ningún atributo no-primario de la tabla es dependiente transitivamente de una clave primaria

Un atributo no-primario es un atributo que no pertenece a ninguna clave candidata. Una *dependencia transitiva* es una dependencia funcional $X \rightarrow Z$ en la cual Z no es inmediatamente dependiente de X, pero sí de un tercer conjunto de atributos Y, que a su vez depende de X. Es decir, $X \rightarrow Z$ por virtud de $X \rightarrow Y$ e $Y \rightarrow Z$.

Una formulación alternativa de la definición de Codd, dada por Carlo Zaniolo[2] en 1982, es ésta: Una tabla está en 3NF si y solo si, para cada una de sus dependencias funcionales $X \rightarrow A$, por lo menos una de las condiciones siguientes se mantiene:

- X contiene A, ó
- X es una superclave, ó
- A es un atributo primario (es decir, A está contenido dentro de una clave candidata)

La definición de Zaniolo tiene la ventaja de dar un claro sentido de la diferencia entre la 3NF y la más rigurosa forma normal de Boyce-Codd (BCNF). La BCNF simplemente elimina la tercera alternativa ("A es un atributo primario").

4.6 Forma normal Boyce-Codd

La Forma Normal de Boyce-Codd (o FNBC) es una forma normal utilizada en la normalización de bases de datos. Es una versión ligeramente más fuerte de la Tercera forma normal (3FN). La forma normal de Boyce-Codd requiere que no existan dependencias funcionales no triviales de los atributos que no sean un conjunto de la clave candidata. En una tabla en 3FN, todos los atributos dependen de una clave, de la clave completa y de ninguna otra cosa excepto de la clave (excluyendo dependencias triviales, como $A \rightarrow A$). Se dice que una tabla está en FNBC si y solo si está en 3FN y cada dependencia funcional no trivial tiene una clave candidata como determinante. En términos menos formales, una tabla está en FNBC si está en 3FN y los únicos determinantes son claves candidatas.

4.7 Algoritmos de descomposición

Ahora se puede exponer un método general para descomponer los esquemas de relación de manera que satisfagan
FNBC. La Figura 7.13 muestra un algoritmo para esta tarea. Si R no está en FNBC se puede descomponer
en un conjunto de esquemas en FNBC, R1, R2,..., Rn utilizando este algoritmo. El algoritmo utiliza las dependencias
(«testigos») que demuestran la violación de FNBC para llevar a cabo la descomposición.

La descomposición que genera este algoritmo no sólo está en FNBC, sino que también es una descomposición
de reunión sin pérdida. Para ver el motivo de que el algoritmo genere sólo descomposiciones de reunión sin pérdida
hay que observar que, cuando se reemplaza el esquema Ri por (Ri – β) y (α, β), se cumple α →β y (Ri – β)
∩ (α, β) = α.

Se aplicará el algoritmo de descomposición FNBC al esquema Esquema-empréstito que se empleó en el Apartado 7.2
como ejemplo de mal diseño de base de datos:

Esquema-empréstito = (nombre-sucursal, ciudad-sucursal, activo, nombre-cliente, número-préstamo, importe)

El conjunto de dependencias funcionales que se exige que se cumplan en Esquema-empréstito es

nombre-sucursal → activo ciudad-sucursal
número-préstamo → importe nombre-sucursal

4.8 Formas normales superiores

Puede que en algunos casos el empleo de las dependencias funcionales para la descomposición de los esquemas no sea suficiente para evitar la repetición innecesaria de información. Considérese una ligera variación de la definición del conjunto de entidades empleado en la que se permite que los empleados tengan varios números de teléfono, alguno de los cuales puede ser compartido entre varios empleados.

Entonces, *numero_telefono* será un atributo multivalorado y, de acuerdo con las reglas para la generación de esquema a partir de los diseños E-R, habrá dos esquemas, uno por cada uno de los atributos multivalorados numero_telefono y nombre_subordinado:
(id_empleado,nombre_subordinado)(id_empleado,numero_telefono)

Si se combinan estos esquemas para obtener
(id_empleado,nombre_subordinado,numero_telefono)

Se descubre que el resultado se halla en la FNBC, ya que solo se cumplen dependencias funcionales no triviales. En consecuencia, se puede pensar que ese tipo de combinación es una buena idea. Sin embargo se trata de una mala idea, como puede verse si se considera el ejemplo de un empleado con dos subordinados y dos números de teléfono. Por ejemplo, sea el empleado con id_empleado 9999999999 que tiene dos subordinados llamados “David” y “Guillermo” y dos números de teléfono, 512555123 y 512555432.

4.9 Integridad de las bases de datos

Integridad de las Bases de Datos, la integridad en una base de datos es la corrección y exactitud de la información contenida. Además de conservar la seguridad en un sistema de bases de datos que permite el acceso a múltiples usuarios en tiempos paralelos.

Las condiciones que garantizan la integridad de los datos pueden ser de dos tipos:
Las restricciones de integridad de usuario: son condiciones específicas de una base de datos concreta; son las que se deben cumplir en una base de datos articular con unos usuarios concretos, pero que no son necesariamente relevantes en otra Base de Datos.
Las reglas de integridad de modelo: son condiciones propias de un modelo de datos, y se deben cumplir en toda base de datos que siga dicho modelo.
Los SGBD deben proporcionar la forma de definir las restricciones de integridad de usuario de una base de datos y una vez definida, debe velar por su cumplimiento. Las reglas de integridad del modelo, en cambio, no se deben definir para cada base de datos concreta, porque se consideran preestablecidas para todas las base de datos de un modelo. Un SGBD de un modelo determinado debe velar por el cumplimiento de las reglas de integridad preestablecidas por su modelo.