TEMA 2. Bases de datos relacionales.

1. Conceptos básicos de las Bases de Datos Relacionales

En su artículo, Codd propuso un modelo simple de datos en el que todos ellos se podían representar en tablas constituidas por filas y columnas. A estas tablas se les dio el nombre matemático de *relaciones*, y por eso el modelo se denominó modelo relacional. Codd también propuso dos lenguajes para manipular los datos en las tablas: el álgebra relacional y el cálculo relacional. Ambos lenguajes soportan la manipulación de los datos sobre la base de operadores lógicos en lugar de los punteros físicos utilizados en los modelos jerárquico y en red.

Al manipular los datos sobre una base conceptual en vez de una base física, Codd introdujo otra innovación revolucionaria. En los sistemas de bases de datos relacionales, los archivos completos de datos se pueden procesar con instrucciones sencillas. Sin embargo, los sistemas tradicionales requieren que los datos se procesen de registro en registro. El enfoque de Codd mejoró enormemente la eficiencia conceptual de la programación de la base de datos.

La manipulación lógica de los datos también hace factible la creación de lenguajes de interrogación más accesibles al usuario no especialista en computación. Aunque es bastante difícil crear un lenguaje que pueda ser utilizado por todas las personas sin considerar su experiencia previa en computación, los lenguajes relacionales de consulta hacen posible el acceso a las bases de datos para un grupo de usuarios cada vez mayor.

La publicación de los artículos de Codd, a principios de los años 70, provocó una conmoción en la actividad de las comunicaciones de desarrollo de los sistemas de investigación y de sistemas comerciales, en la medida en que trabajan para producir un sistema de gestión de bases de datos relacionales. El resultado fue la aparición de sistemas relacionales durante la última mitad de los 70 que soportaban lenguajes como el Structured Query Language (SQL), el Query Language (Quel) y el Query-by-Example (QBE). A medida que las computadoras se hicieron populares durante los años 80, los sistemas relacionales para ellas también estuvieron disponibles.

En 1986, el SQL se adoptó como la norma ANSI para los lenguajes relacionales de bases de datos. Esta norma se actualizó en 1989 y en 1992.

El modelo relacional era un intento de simplificar la estructura de las bases de datos. Eliminaba las estructuras explícitas padre/hijo de la base de datos y en su lugar representaba todos los datos en la base de datos como sencillas tablas fila/columna de valores de datos. En la figura siguiente puede verse una sencilla base de datos relacional.

Tabla ASIGNATURAS

GO	NOMBRE
10000	Tratamiento de datos
1	Análisis estadístico
02	Cálculo numérico
…	…

Tabla AULAS

EDIFICIO	NUMERO	CAPACIDAD
E1	11	100
E1	12	120
E2	11	110
…	…	…

Tabla PROFESORES

CODIGO	NOMBRE	APELLIDOS
H0001	Antonio	García García
H0002	Amparo	Pérez Pérez
H0003	Isabel	Fernández Fernández
…	…	…

Ejemplo de base de datos relacional.

Sin embargo, la definición practica de "¿Qué es una base de datos relacional?" resulta menos clara que la definición matemática recogida en el artículo de Codd de 1970. Por ello el propio Dr. Codd escribió en 1985 un artículo estableciendo doce reglas a seguir por cualquier base de datos que fuera "relacional". Las doce reglas de Codd han sido aceptadas como la definición de un SGBD verdaderamente relacional y se analizaran en el siguiente punto. Sin embargo, una definición más informal es:

Una base de datos relacional es una base de datos en donde todos los datos visibles al usuario están organizados estrictamente como tablas de valores, y en donde todas las operaciones de la base de datos operan sobre estas tablas.

La definición anterior elimina estructuras tales como los punteros incorporados de una base de datos jerárquica o en red. Un SGBD relacional puede representar relaciones padre/hijo, pero éstas se representan estrictamente por los valores contenidos en las tablas de la base de datos.

El principio de organización de una base de datos relacional es la *tabla*, una disposición rectangular fila/columna de los valores de datos. Cada tabla de una base de datos tiene un *nombre de tabla* único que identifica sus contenidos. En realidad, cada usuario puede elegir sus propios nombres de tablas sin preocuparse de los nombres elegidos por otros usuarios.

La estructura fila/columna de una tabla puede verse claramente en la figura que viene a continuación. Cada *fila* horizontal de la tabla PROFESORES representa una única entidad física. Juntas todas las filas de la tabla representan todos los profesores de la universidad. Todos los datos de una fila particular de la tabla se aplican al profesor representado mediante esa fila. Cada fila de una tabla contiene exactamente un valor en cada columna. Una tabla puede contener cualquier número de filas, incluso cero filas, denominándose en tal caso *tabla vacía*. Una tabla vacía sigue teniendo una estructura, impuesta por las columnas, sólo que no contiene datos.

Tabla PROFESORES

CODIGO	NOMBRE	APELLIDOS	CATEGORIA	ANTIGUEDAD
H0001	Antonio	García García	Catedrático	15/02/1983
H0002	Amparo	Pérez Pérez	Ayudante	01/09/1997
H0003	Isabel	Fernández Fernández	Titular	19/04/1991
…	…	…	…	…

Ejemplo de una tabla de una base de datos relacional.

Cada *columna* vertical de la tabla PROFESORES representa un atributo que está almacenado en la base de datos para cada profesor. Por ejemplo la columna CATEGORIA contiene la categoría profesional de cada profesor. Para cada columna de una tabla, todos los valores de esa columna contienen el mismo atributo. El conjunto de valores que una columna puede contener se denomina el *dominio* de la columna. Así, el dominio de la columna NOMBRE es cualquier nombre, mientras que el dominio de la columna CATEGORIA es sólo de tres valores, "Catedrático", "Titular" y "Ayudante" (suponemos que esas son las tres únicas categorías profesionales del personal docente en una universidad).

Cada columna de una tabla tiene un nombre de columna que se escribe generalmente como encabezamiento en la parte superior de la columna. Todas las columnas de una tabla deben tener nombres diferentes, pero no está prohibido que columnas de tablas

diferentes tengan nombres idénticos (véanse las tablas ASIGNATURAS y PROFESORES que hemos puesto como ejemplo de BD relacional). Una tabla tiene como mínimo una columna. No existe un número máximo de columnas en una tabla, estando éste limitado por el producto comercial que se utilice.

Como las filas de una tabla relacional no están ordenadas, no se puede seleccionar una fila específica por su posición en la tabla, sino que debemos usar un identificador, que recordemos son un conjunto de atributos de una entidad que determinan de forma unívoca cada uno de los elementos de dicha entidad. Este identificador suele ser conocido en la terminología de las bases de datos relacionales como *clave primaria*. En la figura siguiente podemos ver dos ejemplos de clave primaria. En uno de ellos la clave primaria esta formada por un solo atributo mientras que en el otro caso esta formada por la combinación de dos atributos.

Tabla PROFESORES

CODIGO	NOMBRE	APELLIDOS
H0001	Antonio	García García
H0002	Amparo	Pérez Pérez
H0003	Isabel	Fernández Fernández
…	…	…

Clave primaria

Tabla AULAS

EDIFICIO	NUMERO	CAPACIDAD
E1	11	100
E1	12	120
E2	11	110
…	…	…

Clave primaria

Dos ejemplos de claves primarias en una base de datos relacional.

Una de las principales diferencias entre el modelo relacional y los modelos de datos anteriores es que los punteros explícitos, tales como las relaciones padre/hijo de una base de datos jerárquica, están prohibidos en las bases de datos relacionales. Obviamente estas relaciones siguen existiendo, pero no mediante punteros explícitos sino mediante *valores de datos comunes* almacenados en cada una de las tablas. Por ejemplo, en la siguiente figura, la tabla ASIGNATURAS contiene la columna PROFESOR, que indica el profesor de la asignatura mediante su clave primaria. Una columna de una tabla cuyo valor coincide con una clave primaria de alguna otra tabla se denomina *clave ajena o foránea*. La clave ajena obviamente estará formada por uno o varios atributos según suceda en la clave primaria con la cual esta relacionada. Todas las relaciones de una base de datos relacional están representadas de este modo.

Tabla ASIGNATURAS

CODIGO	NOMBRE	PROFESOR
10000	Tratamiento de datos	H0001
10001	Análisis estadístico	H0002
10002	Cálculo numérico	H0003
…	…	…

Clave primaria Clave foránea

Tabla PROFESORES

CODIGO	NOMBRE	APELLIDOS
H0001	Antonio	García García
H0002	Amparo	Pérez Pérez
H0003	Isabel	Fernández Fernández
…	…	…

Clave primaria

Ejemplo de clave ajena en una base de datos relacional.

Para concluir este apartado, debemos mencionar que todos estos desarrollos hicieron avanzar enormemente el estado del arte en los temas de gestión de bases de datos y aumentaron la disponibilidad de información en las bases de datos colectivas. Por tanto, el enfoque relacional ha resultado bastante ventajoso.

Actualmente, los sistemas relacionales son un estándar en el mercado, especialmente en operaciones comerciales. Naturalmente, tantos los sistemas orientados a archivos, como también los sistemas de bases de datos jerárquicos y en redes son todavía abundantes y, para ciertas aplicaciones, constituyen la solución más eficiente en función de los costes. Sin embargo, durante algún tiempo, la tendencia clara de las compañías ha sido migrar a los sistemas relacionales siempre que fuera posible.

Aún así, sería un error asumir que los sistemas de bases de datos relacionales, ahora disponibles, representan la última palabra en el desarrollo de los SGBD. Los sistemas relacionales de hoy aún están evolucionando y, en algunos aspectos significativos, cambiando su naturaleza subyacente para permitir a los usuarios plantear problemas más complejos. Desde nuestro punto de vista, uno de los cambios más importantes está ocurriendo en el área de las bases de datos orientadas a objetos en las que los datos se representan mediante objetos, que contienen variables y métodos, y su manipulación se realiza mediante mensajes. Un desarrollo adicional de gran importancia es la aparición de la plataforma cliente/servidor como la base para los cálculos y el acceso a las bases de datos en una organización.

2. Las doce reglas de Codd de definición de un SGBD relacional

En un artículo de 1985 publicado en Computerworld, el Dr. Codd presentó doce reglas que una base de datos debe obedecer para que sea considerada relacional. Las doce reglas de Codd se han convertido en la definición teórica de una base de datos relacional. Estas se derivan del trabajo teórico de Codd sobre el modelo relacional y representan realmente más un objetivo ideal que una definición de una base de datos relacional. Dichas doce reglas son:

1. *Regla de información.* Toda la información de una base de datos relacional está representada explícitamente a nivel lógico y exactamente de un modo: Mediante valores en tablas.

2. *Regla de acceso garantizado.* Todos y cada uno de los datos de una base de datos relacional se garantiza que sean lógicamente accesibles recurriendo a una combinación de nombre de tabla, valor de clave primaria y nombre de columna.

3. *Tratamiento sistemático de valores nulo.* Los valores nulos (distinto de la cadena de caracteres vacía o de una cadena de caracteres en blanco y distinta del cero o de cualquier otro número) se soportan en los SGBD completamente relaciones para representar la falta de información y la información inaplicable de un modo sistemático e independiente del tipo de datos.

4. *Catálogo en línea dinámico basado en el modelo relacional.* La descripción de la base de datos se representa a nivel lógico del mismo modo que los datos ordinarios, de modo que los usuarios autorizados puedan aplicar a su interrogación el mismo lenguaje relacional que aplican a los datos regulares.

5. *Regla de sublenguaje completo de datos.* Un sistema relacional puede soportar varios lenguajes y varios modos de uso terminal (por ejemplo, el modo de rellenar con blancos). Sin embargo, debe haber al menos un lenguaje cuyas sentencias sean expresables mediante alguna sintaxis bien definida, como cadenas de caracteres, y que sea completa en cuanto al soporte de todos los puntos siguientes:

- Definición de datos.
- Definición de vista.
- Manipulación de datos (interactiva y por programa).
- Restricciones de integridad.
- Autorización.
- Fronteras de transacciones (comienzo, cumplimiento y vuelta atrás).

6. *Regla de actualización de vista.* Todas las vistas que sean teóricamente actualizables son también actualizables por el sistema.

7. *Inserción, actualización y supresión de alto nivel.* La capacidad de manejar una relación de base de datos o una relación derivada como un único operando se aplica, no solamente a la recuperación de datos, sino también a la inserción, actualización y supresión de los datos.

8. *Independencia física de los datos.* Los programas de aplicación y las actividades terminales permanecen lógicamente inalterados cualquiera que sean los cambios efectuados, ya sea a las representaciones de almacenamiento o a los métodos de acceso.

9. *Independencia lógica de los datos.* Los programas de aplicación y las actividades terminales permanecen lógicamente inalterados cuando se efectúen, sobre las tablas de base, cambios preservadores de la información de cualquier tipo que teóricamente permita alteraciones.

10. *Independencia de integridad.* Las restricciones de integridad específicas para una base de datos relacional particular deben ser definibles en el sublenguaje de datos relacional y almacenables en el catálogo, no en los programas de aplicación.

11. *Independencia de distribución.* Un SGBD relacional tiene independencia de distribución.

12. *Regla de no subversión.* Si un sistema relacional tiene un lenguaje de bajo nivel (un solo registro a la vez), ese bajo nivel no puede ser utilizado para subvertir o suprimir las reglas de integridad y las restricciones expresadas en el lenguaje relacional de nivel superior (múltiples registros a la vez).

La regla 1 es la definición informal de una base de datos relacional. La regla 2 refuerza la importancia de las claves primarias para localizar datos en la base de datos. El nombre de la tabla localiza la tabla correcta, el nombre de la columna encuentra la columna correcta y el valor de la clave primaria encuentra la fila que contiene un dato individual de interés. La regla 3 requiere soporte para falta de datos mediante el uso de valores NULL.

La regla 4 requiere que una base de datos relacional sea autodescriptiva. En otras palabras, la base de datos debe contener ciertas *tablas de sistema* cuyas columnas describan la estructura de la propia base de datos.

La regla 5 ordena la utilización de un lenguaje de base de datos relacional. El lenguaje debe ser capaz de soportar todas las funciones básicas de un SGBD (creación de una

base de datos, recuperación y entrada de datos, implementación de la seguridad de la base de datos, etc.).

La regla 6 trata de las vistas, que son tablas virtuales utilizadas para dar, a diferentes usuarios de una base de datos, diferentes vistas de su estructura. Es una de las reglas más difíciles de implementar en la práctica.

La regla 7 refuerza la naturaleza orientada a conjuntos de una base de datos relacional. Requiere que las filas sean tratadas como conjuntos en operaciones de inserción, supresión y actualización. La regla esta diseñada para impedir implementaciones que sólo soportan la modificación o recorrido fila a fila de la base de datos.

La regla 8 y la regla 9 aíslan al usuario o al programa de aplicación de la implementación de bajo nivel de la base de datos. Especifican que las técnicas específicas de acceso a almacenamiento utilizadas por el SGBD, e incluso los cambios a la estructura de las tablas en la base de datos, no deberían afectar a la capacidad del usuario de trabajar con los datos.

La regla 10 dice que el lenguaje de base de datos debería soportar las restricciones de integridad que restringen los datos que pueden ser introducidos en la base de datos y las modificaciones que pueden ser efectuadas en ésta.

La regla 11 dice que el lenguaje de base de datos debe ser capaz de manipular datos distribuidos, es decir, localizados en otros sistemas informáticos. Por último, la regla 12 impide “otros caminos” en la base de datos que pudieran subvertir su estructura relacional y su integridad.

Ningún SGBD relacional actualmente disponible satisface totalmente las doce reglas de Codd. De hecho, se elaboran pruebas para productos SGBD comerciales, que muestran lo bien o mal que éstos satisfacen cada una de las reglas.

3. Principios de diseño de las B.D. relacionales

3.1. Introducción a las metodologías de diseño

Un *sistema de información* (SI) es un conjunto de elementos que funcionan conjuntamente con el objetivo de recoger, tratar, manipular y aportar la informaciones necesarias para el desarrollo de las actividades de una empresa u organización. Un SI puede incluir procesos manuales o automáticos.

Uno de los elementos principales de un SI es la base de datos (BD). Las BD son ejemplos típicos de grandes sistemas de software con tres características importantes:

- Hay una gran cantidad de datos que deben ser almacenados en memoria externa y que deben ser organizados de forma que los datos elementales puedan ser recuperados y actualizados fácil y eficientemente.

- Los datos guardan entre sí complejas interrelaciones. La información incluye restricciones estáticas y dinámicas, como los valores permitidos o las posibles evoluciones.

- Los datos deben ser compartidos entre diferentes usuarios y el sistema debe mantener la integridad de la información.

Un *modelo* es una representación de un sistema que pretende simplificar su comprensión poniendo en evidencia ciertos aspectos del sistema mientras otros son ocultados. Los modelos se utilizan para facilitar la tarea de diseño de los SI complejos, ya que facilitan 'pensar en lo que se está haciendo' y permiten comprobar si los resultados se adecuan al problema y en caso contrario, corregirlos.

Los modelos pueden tener distintos niveles de abstracción. En los SI se utilizan tres tipos de modelos con diferentes niveles de abstracción:

- El *modelo físico*, que describe completamente el sistema: circulación y tratamiento de la información, elementos informáticos y elementos manuales. Para la BD el modelo físico representa la organización de la información sobre los soportes de almacenamiento.

- El *modelo lógico*, que describe las informaciones y las manipulaciones a que son sometidas. Este modelo hace abstracción de los soportes materiales de almacenamiento. El modelo lógico sobre una BD representa la definición de la información sobre el SGBD elegido para el desarrollo del SI.

- El *modelo conceptual*, que describe el contenido subyacente al modelo lógico, esto es, el significado de las informaciones y las relaciones que las unen. Este modelo hace abstracción de las manipulaciones de la información.

En los siguientes temas veremos más detalladamente los conceptos y las técnicas necesarias para diseñar bases de datos pero antes debemos ver cuál es la metodología a seguir.

3.2. Metodologías de diseño de bases de datos

Las dificultades inherentes al diseño de una base de datos han de afrontarse con procedimientos ordenados y metódicos. Existen distintas metodologías para el diseño de SI y de BD que identifican las etapas del diseño, lo que podemos obtener en cada una de estas etapas, así como las herramientas necesarias para desarrollar esta actividad. Algunas de las metodologías más utilizadas son MERISE y SSADM. A continuación describiremos una de las metodología para el desarrollo de BD relacionales que sigue las recomendaciones más comunes encontradas en la literatura.

i) Fase de análisis de requisitos de usuario

El objetivo de esta etapa es describir con precisión el contenido de información de la base de datos y determinar las demandas de transacción que sufrirá el sistema.

Las especificaciones del sistema pueden describirse informalmente utilizando descripciones narrativas, o formalmente utilizando un modelo de datos. El modelo de datos a utilizar debe ser suficientemente 'expresivo' como para poder representar toda la información de la BD.

Al integrar la BD las necesidades de información de distintos grupos de usuarios y de distintas aplicaciones dentro de la empresa u organización, será un previo identificar

todos los grupos de usuarios que interactúan con la BD. Posteriormente se obtendrá para cada uno de estos grupos la descripción detallada de sus requisitos. Cada grupo de estas especificaciones constituye una vista particular de la BD.

En esta etapa se identificarán los objetos o eventos del mundo real que almacenará la BD, sus propiedades y las relaciones que mantienen entre ellos. Se identificarán las condiciones de integridad de la información que darán lugar a restricciones de manipulación. Se identificarán las autorizaciones de acceso a la información por parte de los distintos grupos de usuarios. También puede ser interesante en esta etapa tener una estimación del volumen de datos a almacenar.

Por último, se identificarán las operaciones a realizar sobre la información. Esta información es relevante en el modelo relacional únicamente para el diseño del modelo físico de la BD. Por esta razón, en algunos casos no se requiere. Se describirán la naturaleza de las transacciones, la frecuencia con que se realizan, la información que utilizan y producen, así como el flujo de información entre ellas.

ii) Fase de diseño del esquema conceptual

En esta etapa se combinan los requisitos de los distintos grupos de usuarios del sistema para contribuir a una descripción única y coherente de toda la información a almacenar en la BD. El esquema conceptual se desarrolla en un modelo semántico y define las entidades en la BD, las relaciones entre ellas, las restricciones de integridad de la información, los grupos de usuarios y las autorizaciones de acceso de cada uno de ellos.

Este proceso se conoce a veces como integración de vistas, y en el se produce una descripción global de la BD eliminando la redundancia e inconsistencia entre las especificaciones de distintos grupos de usuarios. El modelador debe reconocer entre las diferentes vistas los sinónimos y homónimos y aquellos tipos de datos que pertenecen a las mismas categorías.

En esta etapa se construye un modelo de datos que describe cada uno de los objetos y sus relaciones. Se identifican los atributos que forman las claves de acceso a los objetos, se decide la representación de las interrelaciones y se identifican las propiedades opcionales y fijas, para lo cual se utiliza un modelo semántico de datos, comúnmente el modelo Entidad/Relación, que será el que utilicemos nosotros en este curso.

iii) Fase de diseño lógico

Durante esta etapa, el modelo conceptual se transforma en el modelo empleado por el SGBD donde se implementará el sistema. El modelo que más se utiliza hoy por hoy es el modelo relacional, aunque se utilizan todavía el modelo de red y el jerárquico.

Para BD relacionales, en esta etapa se construyen las tablas a partir de los elementos del esquema conceptual, incluyendo las relaciones entre entidades y las reglas de integridad. En esta etapa se crean los usuarios y se administran las autorizaciones para acceder a la información.

El modelo E/R desarrollado en el paso anterior puede transformarse directamente en tablas, siguiendo una serie de reglas de transformación. La mayoría de modelos ofrece estas reglas para una transformación semiautomática del esquema conceptual de relaciones de la BD.

iv) Fase de normalización de los esquemas relacionales

En muchos casos, las tablas resultantes del proceso anterior se someten a un proceso posterior de normalización que elimina redundancias de la información no eliminadas y que repercuten en dificultades en el procesamiento de la información.

v) Fase de diseño físico

En función de las operaciones a realizar sobre la BD, se definen los mecanismos de almacenamiento y organización de la información, incluyendo la creación de índices o la agrupación en 'clusters' de los datos. En algunos casos puede ser necesario desnormalizar la información (añadiendo redundancia) para conseguir el rendimiento deseado de la aplicación.

vi) Fase de implementación

Es la transformación del modelo de datos y los diseños realizados en una base de datos en funcionamiento, que opere en un determinado equipo y bajo el control de un SGBD.

vii) Fase de test

La etapa de test tiene el propósito de descubrir errores aparecidos en las etapas anteriores y que provocan un comportamiento del sistema diferente al previsto. Es también objetivo de la etapa de test el comprobar junto a los usuarios que el sistema satisface los objetivos establecidos durante la etapa de especificación.

viii) Fase de mantenimiento

La fase de mantenimiento incluye la corrección de errores que pudieran aparecer durante la etapa de operación del sistema. La implementación también comprende las modificaciones que el usuario pudiera solicitar a partir de la experiencia de operación del sistema o de nuevas necesidades, la mejora de la eficacia del sistema y la mejora de los interfaces de usuario.