



UD1

DISEÑO CONCEPTUAL DE BASES DE DATOS

MP_0484
Bases de Datos

1.4 Modelo
relacional

Introducción

El modelo más utilizado en la historia de las bases de datos y actualmente en uso es el modelo relacional. A partir de este momento, el contenido de este tema se basa casi en su totalidad en el modelo relacional. Hasta el momento hemos hablado del nivel conceptual y hemos plasmado gráficamente el enunciado por representar a través del diagrama E/R, este diagrama serviría para cualquier modelo de base de datos (BD). A partir de este punto, comenzaremos a trabajar con la representación aplicada al modelo relacional.

El objetivo pasa por representar gráficamente los enunciados propuestos a tratar a través del modelo relacional, para luego construir nuestra base de datos relacional. Vamos a poder construir el modelo relacional de dos formas: una de ellas es transformando el modelo E/R a relacional y la otra es aplicar el proceso de normalización.

Modelo relacional

A finales de los años 60, Edgar Frank Codd sentó las bases del modelo relacional, que revolucionaría el campo de la informática. En 1970, publicó el influyente documento «A Relational Model of Data for Large Shared Data Banks», donde establece los fundamentos del modelo relacional de bases de datos.

Antes de esto, el paradigma predominante era el modelo en red de bases de datos, representado principalmente por Codasyl. Basándose en los trabajos matemáticos de Cantor y Childs, quienes desarrollaron la teoría de conjuntos y el álgebra relacional, Codd propuso que los datos se organizaran en relaciones, ahora comúnmente conocidas como tablas. Este concepto permitía agrupar datos relacionados con una entidad específica de manera independiente de su almacenamiento físico.

El objetivo principal de Codd era simplificar el uso de las bases de datos, liberando a los usuarios de la necesidad de comprender los detalles internos del sistema. Esta visión innovadora encontró resistencia en IBM, donde trabajaba Codd, ya que la empresa continuaba enfocada en su modelo en red IMS. Sin embargo, otras compañías, especialmente Oracle, adoptaron sus ideas y las implementaron con éxito. Con el tiempo, el modelo relacional ganó popularidad y se convirtió en el estándar de facto en el diseño de bases de datos. Hoy en día, un porcentaje muy alto de las bases de datos siguen este enfoque, gracias a la visión pionera de Codd y a su capacidad para anticipar las necesidades de los usuarios y simplificar su experiencia.

Estructura de las bases de datos relacional

Relación o tabla

Según el modelo relacional propuesto por Codd, el elemento fundamental es lo que se conoce como relación, aunque comúnmente se le denomina tabla. Codd definió las relaciones utilizando un lenguaje matemático, pero se pueden asociar fácilmente a la idea de una tabla con filas y columnas para una comprensión más sencilla.

Es importante no confundir la noción de relación según el modelo de Codd con lo que representa una relación en el modelo Entidad/Relación de Chen, ya que son conceptos distintos. Las relaciones en el modelo relacional constan de dos componentes principales:

- **Atributos:** se refiere a cada una de las propiedades de los datos que se almacenan en la relación, como el nombre, el número de identificación, etc.
- **Tuplas:** se refiere a cada elemento individual de la relación. Por ejemplo, si una relación almacena información sobre personas, una tupla representaría a una persona concreta.

En términos de representación visual, como una relación se muestra como una tabla, es común entender que las columnas de la tabla corresponden a los atributos, mientras que las filas representan las tuplas individuales que contienen los valores de esos atributos para cada entidad en la base de datos (figura 1).

Atributo 1	Atributo 2	Atributo 3		Atributo n
Valor 1,1	Valor 1,2	Valor 1,3	Valor 1,n
← Tupla →				
Valor 2,1	Valor 2,2	Valor 2,3	Valor 2,n
.....
Valor m,1	Valor m,2	Valor m,3	Valor m,n

Figura 1. Estructura de una relación

La relación cuenta con seis propiedades que siempre han de cumplirse:

- Tiene un nombre único.
- Cada atributo toma un solo valor por tupla
- Cada atributo tiene un nombre distinto en cada relación
- Cada tupla es única
- El orden de las tuplas es irrelevante
- El orden de los atributos es irrelevante

Tupla

Cada fila de la relación, también conocida como tupla, se asocia con la noción tradicional de registro. Una tupla representa cada una de las filas de la tabla. Es fundamental que cada tupla satisfaga dos criterios clave:

- Cada tupla debe representar un elemento concreto del mundo real.
- No puede existir duplicación de tuplas, es decir, no puede haber dos tuplas idénticas con todos sus valores coincidentes.

Dominio

Un dominio contiene todos los posibles valores que un determinado atributo puede tomar y dos atributos distintos pueden compartir el mismo dominio. En esencia, un dominio es un conjunto finito de valores del mismo tipo. Existen dos formas de definir el contenido de un dominio:

- **Intensión:** se define indicando la definición exacta de sus posibles valores. Por ejemplo, el dominio de edades de los niños de primaria podría definirse como números enteros entre 3 y 6, indicando que un niño solo puede tener una edad entre 3 y 6 años.
- **Extensión:** se enumeran algunos valores del dominio y se sobreentienden los restantes a partir de estos. Por ejemplo, el dominio de provincias podría definirse por extensión como: Burgos, Valladolid, Vizcaya, etc.

Además, los dominios pueden clasificarse en:

- **Generales:** los valores están comprendidos entre un máximo y un mínimo.
- **Restringidos:** solo pueden tomar un conjunto específico de valores.

Grado

Hace referencia a el número de columnas (atributos) de la relación o tabla. Cabe destacar que cuanto mayor es el grado de una relación, mayor es su complejidad al manejarla.

Cardinalidad

Será el número de tuplas de la relación, el número de filas de la tabla.

Clave

Existen varios tipos de claves que tienen el mismo significado que hemos visto en campo clave del diagrama entidad/relación:

- **Clave candidata:** es un conjunto de uno o más atributos (columnas) que puede identificar de manera única una fila en una tabla. En otras palabras, cada clave candidata tiene la capacidad de actuar como una clave primaria debido a su propiedad de unicidad y no nulidad.
- **Clave primaria:** un campo o conjunto de campos que identifica de manera única a cada fila de una tabla. Es un concepto fundamental en el diseño de bases de datos relacionales, donde garantiza que no puede haber dos filas con el mismo valor en la clave primaria. Algunas de las características de las claves primarias son:
 - Unicidad.
 - No permite nulos.
 - Indexación automática.
 - Identificación única.
- **Clave alternativa:** de todas las claves posibles se denominan claves candidatas, una de ellas es la clave principal y el resto son claves alternativas.
- **Clave foránea, ajena o externa:** son aquel campo o campos que sin clave principal es clave en otras tablas. Son las claves que nos van a permitir relacionar unos campos de unas tablas con otros.

Tabla Productos

<u>CodProducto</u>	Nombre	Precio	CodProveedor
C025	Manzanas	2,5 €	P001
C026	Peras	3,5 €	P002
C27	Limonos	1,5 €	P001

Tabla Proveedores

<u>CodProv</u>	DNI_CIF	NombreProv	DirecciónProv	FechaContrato
P001	30526827F	Almacenes Ruíz	Xxxx	15/02/2021
P002	B41414141	Alimentación Fernández	AAAAAA	17/02/2020
P003	58585858X	Suministros RESA	Cccccc	21/02/2021

Figura 2. Ejemplo: tabla proveedores y productos

Como se puede ver en la figura 2, en la tabla Productos, la clave principal es el CodProducto y CodProveedor sería clave ajena. En la tabla Proveedores, las claves candidatas son CodProv y DNI_CIF, de las cuales la clave principal es CodProv y clave alternativa DNI_CIF.

Nomenclaturas posibles del modelo relacional

En la figura 3, se puede ver una correspondencia entre los términos a utilizar entre los distintos sistemas de representación.

Asociación de términos		
Nomenclatura relacional	Nomenclatura tabla	Nomenclatura ficheros
Relación	Tabla	Fichero
Tupla	Fila	Registro
Atributo	Columna	Campo
Grado	N.º columnas	N.º campos
Cardinalidad	N.º filas	N.º registros

Figura 3. Correspondencia entre nombres

Tratamiento de nulos

El valor nulo se utiliza para indicar que un identificador (un atributo, un campo u objeto) no tiene contenido. Cuando en programación un puntero apunta a null, se dice que no está apuntando a ningún objeto. En las bases de datos relacionales, el valor nulo (null) ofrece más posibilidades, aunque su significado básico de valor vacío no cambia. Emplear null en claves ajenas, indica que el registro actual no está relacionado con ningún otro registro.

En otros atributos, un valor nulo señala que la tupla correspondiente carece de ese atributo específico. Por ejemplo, en una tabla de personas, un valor nulo en el atributo teléfono indicaría que esa persona no tiene teléfono. En la tabla Productos vista anteriormente (figura 2), un producto sin CodProveedor, significaría que ese producto no este asignado a ningún proveedor.

Restricciones del modelo relacional

Se trata de condiciones que deben cumplir las tuplas (filas) de las relaciones (tablas) de la base de datos. Son las siguientes:

- **Del modelo:** son aquellas que son inherentes al modelo relacional, se han visto con anterioridad como características del modelo:
 - No puede haber dos tuplas iguales.
 - El orden de las tuplas no es significativo.
 - El orden de los atributos no es significativo.
 - Cada atributo solo puede tomar un valor en el dominio en el que está inscrito.
 - Clave primaria, primary key. Es obligatoria, como hemos visto antes al trabajar los tipos de atributos, marca uno o más atributos como identificador de la tupla. De esta forma, esos atributos que son primary key no podrán repetir valores ni tampoco ser nulos.
- **Semánticas:** son las que se pueden incorporar a los datos, ya que así se representa con mayor precisión y control la realidad del enunciado por tratar:
 - **Unicidad, unique:** impide que los valores de los atributos marcados de esta forma puedan repetirse. Es la restricción que se pone a las claves alternativas.
 - **Obligatoriedad, not null:** no permite que el atributo definido así pueda quedar vacío.
 - **Regla de validación, check:** condición lógica que debe cumplir un atributo para ser válido. Por ejemplo, que la edad sea un número positivo, la fecha de inicio sea menor que la fecha de fin, los temas de un libro sean: literatura, ciencias, etc.
 - **Disparadores, triggers:** Se trata de pequeños programas almacenados en la BD que realizan una determinada acción. Por ejemplo, cuando se añade una tupla, se actualice un dato; cuando se vende un producto, tres botellas de aceite, a un cliente, se actualice restando tres en el stock del aceite en el almacén.
 - **Integridad referencial, foreign key:** sirve para establecer las claves ajenas (secundarias, externas, foráneas). El SGBD vela por la integridad de los datos almacenados, si intentamos modificar o borrar valores de una clave principal asociada con claves ajenas de otras tablas, no nos lo va a permitir.