

Capítulo 3.- El Modelo Relacional

- 1.- INTRODUCCIÓN
- 2.- TERMINOLOGÍA DEL MODELO RELACIONAL
- 3.- PASO DEL MODELO ENTIDAD/RELACIÓN AL MODELO RELACIONAL
- 4.- EJERCICIOS

1.- INTRODUCCIÓN

En 1970 Codd publicó un trabajo proponiendo un nuevo modelo de datos llamado relacional. Está basado en conceptos muy sencillos teniendo asociada la teoría de normalización de relaciones que tiene por objeto la eliminación de los comportamientos anómalos de las relaciones durante los procesos de manejo de la información

El modelo relacional persigue los siguientes **objetivos**:

a.- **Independencia física**: El modo cómo se almacenan los datos no debe influir en su manipulación lógica y, por tanto, los usuarios que acceden a esos datos no han de modificar sus programas por cambios en el almacenamiento físico.

b.- **Independencia lógica**: Añadir, eliminar o modificar cualquier elemento de la BD no debe repercutir en los programas y/o usuarios que están accediendo a subconjuntos parciales de los mismos (vistas).

c.- **Flexibilidad**: Ofrecer a cada usuario los datos de la forma más adecuada a la correspondiente aplicación.

d.- **Uniformidad**: Las estructuras lógicas de los datos presentan un aspecto uniforme (tablas), lo que facilita la concepción y manipulación de la BD por parte de los usuarios.

e.- **Sencillez**: Las características anteriores, así como unos lenguajes de usuario muy sencillos, producen como resultado que el modelo relacional (MR) sea fácil de comprender y de utilizar por parte del usuario final.

Propone una representación de la información que:

i.- Origine esquemas que representen fielmente la información, los objetos y relaciones entre ellos existentes en el dominio del problema.

ii.- Pueda ser entendida fácilmente por los usuarios que no tienen una preparación previa en éste área.

iii.- Haga posible ampliar el esquema de la base de datos sin modificar la estructura lógica existente y, por tanto, sin modificar los programas de aplicación.

iv.- Permita la máxima flexibilidad en la formulación de los interrogantes previstos, y no previstos, sobre la información mantenida en la Base de Datos.

2.- TERMINOLOGÍA DEL MODELO RELACIONAL

La estructura básica del Modelo es la RELACIÓN y se puede representar en forma de tabla de dos dimensiones.

Una relación tiene un nombre, un conjunto de atributos que representan sus propiedades y un conjunto de tuplas que incluyen los valores que cada uno de los atributos toma para cada elemento de la relación.

Como se ha dicho, una relación se representa mediante una tabla bidimensional (las columnas son los atributos de la relación y las filas son las tuplas).

La relación debe cumplir las siguientes condiciones:

- a.- Debe tener un solo tipo de fila, cuyo formato queda definido por el esquema de la relación, por lo tanto todas las filas tienen las mismas columnas.
- b.- Cada fila debe ser única y no pueden existir filas duplicadas.
- c.- Cada columna debe ser única y no pueden existir columnas duplicadas.
- d.- Cada columna debe estar identificada por un nombre específico.
- e.- El valor de una columna para una fila debe ser único, así, no pueden existir múltiples valores en una posición de una columna.
- f.- Los valores de una columna deben al dominio que representa, y es posible que un mismo dominio se utilice para definir los valores de varias columnas.

El DOMINIO es el conjunto de todos los posibles valores para una o más columnas (o atributos) de una relación. Por tanto, los valores contenidos en un atributo (o columna) deben estar entre un máximo y mínimo predefinido por el dominio.

Pueden distinguirse dos tipos de dominios diferentes:

a.- **Dominios generales o continuos:** aquellos que contienen todos los posibles valores entre un máximo y un mínimo predefinido. Ejemplos:

DNI: todos los números enteros y positivos de 8 dígitos.

Peso de material: todos los números reales y positivos.

Saldo de una cuenta: todos los números reales, positivos y negativos.

b.- **Dominios restringidos o discretos:** aquellos que contienen ciertos valores específicos entre un máximo y un mínimo predefinido. Ejemplos:

Estado civil: compuesto por soltero, casado, viudo y divorciado.

Provincia: una de las provincias españolas.

En una relación, a las filas se le denomina TUPLAS, y a las columnas ATRIBUTOS. Dicho de un modo formal, un atributo es la interpretación de un determinado dominio en una relación, es decir, el papel que juega en la misma.

Un atributo y un dominio pueden llamarse igual, pero:

a.- Un atributo está siempre asociado a una relación, mientras que un dominio tiene existencia propia con independencia de las relaciones.

b.- Un atributo representa una propiedad de una relación.

c.- Un atributo toma valores de un dominio.

d.- Varios atributos distintos (de la misma o de diferentes relaciones) pueden tomar sus valores del mismo dominio.

Se denomina GRADO al número de atributos que la forma.

Se llama CARDINALIDAD al número de tuplas que contiene.

Atributo 1	Atributo 2	Atributo N
xxxxxxxxxx	Xxxxxxxxxx	Xxxxxxxxxx
Xxxxxxxxxx	Xxxxxxxxxx	Xxxxxxxxxx
Xxxxxxxxxx	Xxxxxxxxxx	Xxxxxxxxxx
Xxxxxxxxxx	Xxxxxxxxxx	Xxxxxxxxxx

Esta sería una relación con un determinado Nombre.

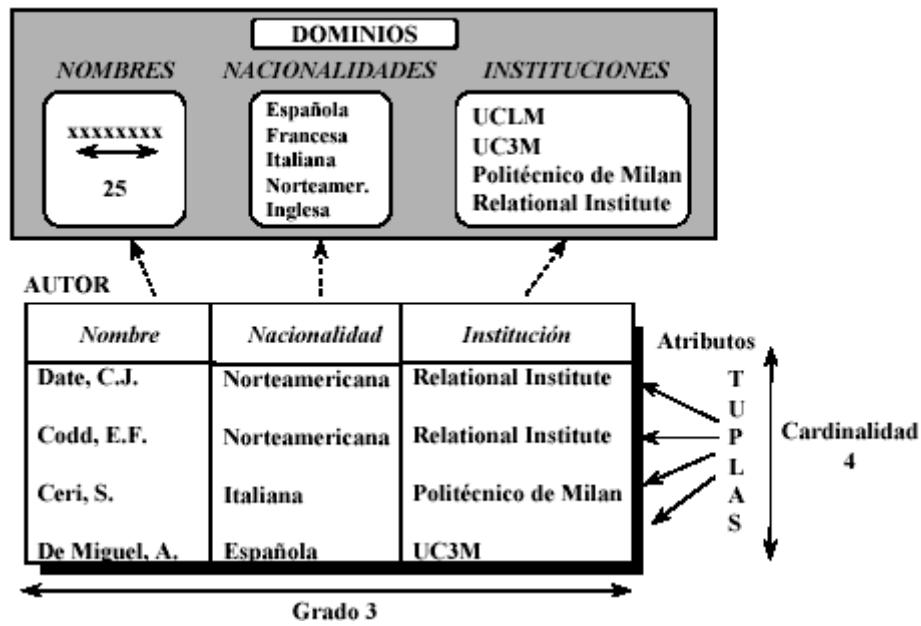
Cada una de las filas (por debajo de Atributo i) se llamará tupla.

Ejemplo:

COD_MATRICULA	NOMBRE	CIUDAD	COD_GRUPO
101	José	Salamanca	11
102	Pepe	Ciudad Real	11
203	Maria	Madrid	23
500	Ignacio	Barcelona	54
54	Leticia	Munich	--

El ejemplo muestra una relación llamada ALUMNO, que consta de los atributos Cod_matrícula, Nombre, Ciudad y Cod_grupo (nombres de las columnas de la tabla). Cada fila de la tabla muestra una tupla correspondiente a los datos de un alumno.

El dominio de Cod_matricula sería "códigos" formado por un conjunto de cadenas de longitud 3 caracteres.



Comparación de la terminología relación, tabla, fichero

RELACIÓN	~	TABLA	~	FICHERO
TUPLA		FILA		REGISTRO
ATRIBUTO		COLUMNA		CAMPO
GRADO		Nº DE COLUMNAS		Nº DE CAMPOS
CARDINALIDAD		Nº DE FILAS		Nº DE REGISTROS

Modelo

Relacional

(teoría)

SGBD

Relacionales

(implementación)

Sistemas

de Ficheros

Clásicos

El modelo relacional debe satisfacer una serie de reglas de integridad mediante las cuales se garantiza la consistencia de la información que pueda ser manejada:

- En una relación no puede haber 2 tuplas iguales (obligatoriedad de clave primaria).
- El orden de las tuplas y el de atributos no es relevante. (Aunque es recomendable que las tuplas estén en orden tomando como referencia un atributo, normalmente el primero, que deberá coincidir con lo que llamamos clave de la relación).
- Cada atributo sólo puede tomar un único valor del dominio sobre el cual está definido.
- Ningún atributo que forme parte de la clave primaria de una relación puede tomar un valor nulo (regla de integridad de entidad o de clave).
- Se pueden imponer , en teoría, otra serie de restricciones que garanticen la integridad del modelo y por lo tanto de la información almacenada en la BdD, restricciones concernientes con:
 - - a) Los valores permitidos para los atributos que forman parte de las relaciones existentes en el esquema. Por ejemplo, valor máximo y mínimo, lista de valores, etc...
 - b) Condiciones que determinan el valor que pueden tomar los atributos. Estas condiciones pueden definirse en base a diferentes predicados: en función del valor de otros atributos de la misma o diferente relación, o al estado de la base de datos, o en base al usuario, etc...

CLAVES.

En una relación, es necesario poder determinar una tupla concreta, lo cual es posible mediante la clave.

Clave es un atributo o conjunto de atributos cuyos valores distinguen unívocamente una tupla específica de una tabla o relación.

Cada una de las condiciones de una tabla relacional o relación es que no pueden existir filas duplicadas, ello implica que siempre tiene que existir al menos una clave, que en el peor de los casos estará formada por todos los atributos. La clave se debe buscar a través de todos los posibles valores (dominio) de los atributos, y no entre los valores de unas tuplas concretas.

Si una tabla dispone de varias claves, a éstas se le denominan CLAVES CANDIDATAS, puesto que más adelante entre todas ellas tendremos que elegir una que será la que por excelencia identifique la tupla. A esta clave se le denomina CLAVE PRINCIPAL o primaria y al resto CLAVES ALTERNATIVAS o secundarias. Los atributos que pertenecen a la clave primaria se denominan atributos primarios y el resto atributos no primarios o secundarios.

Ejemplo:

DNI	NOMBRE	DIRECCIÓN	CIUDAD
493	Pedro	C/ Col, 9	Jaén
181	Luis	C/ Pez, 7	Madrid
830	María	C/ Paz, 12	Madrid
341	Ana	Pza. Mayor, 5	Salamanca
679	Juan	C/ Sol, 5	Madrid
911	Pilar	C/ Mar, 51	Sevilla

"PERSONAL"

DNI	N_MATRIC	MARCA	MODELO	COLOR	KM
723	M-3443	Ford	Fiesta	Blanco	579
181	M-5996	Seat	Ibiza	Rojo	39901
524	M-4283	Opel	Corsa	Blanco	34567
524	M-2103	Opel	Corsa	Verde	21345
181	M-3911	BMW	318	Gris	42354
911	SE-6720	Renault	R-21	Rojo	34521
308	M-4283	Opel	Corsa	Blanco	34567

"COCHES-PERSONAL"

En personal la clave es DNI, puesto que cualquier valor del dominio de dicho atributo identifica a una única tupla.

En coches-personal se representan coches que pueden disponer un individuo determinado. Un coche que se identifica por el N_Matrícula puede estar compartida por varios individuos, por lo tanto, el N_Matrícula no identifica a una tupla determinada puesto que el coche se puede encontrar en distintos individuos (véase el coche del DNI 524 y 308).

El DNI tampoco nos sirve para localizar una única tupla puesto que un individuo puede disponer de más de un coche (ej. DNI 524). Lógicamente tampoco serán los demás ya que distintos coches pueden tener la misma marca, el mismo modelo, el color o incluso los mismos km.

Así, la clave debe estar formada por más de un atributo y será DNI junto con N_Matrícula. Esto quiere decir que de todas las tuplas que tenga la tabla se distinguirán entre si mediante los valores concatenados del DNI y N_Matrícula o lo que es igual, si conocemos el valor de un DNI y de un número de matricula asociado, identificaremos a una única tupla.

Cuando vayamos a analizar la clave principal, tendremos en cuenta sobre su dominio las siguientes consideraciones:

- ⇒ Sus valores siempre deben ser conocidos (diferentes de nulo).
- ⇒ La memoria que ocupen debe ser mínima
- ⇒ Su codificación ser sencilla
- ⇒ El contenido de sus valores no debe variar
- ⇒ Que se utilicen en otras tablas para crear una interrelación (mediante lo que se llama claves ajenas).

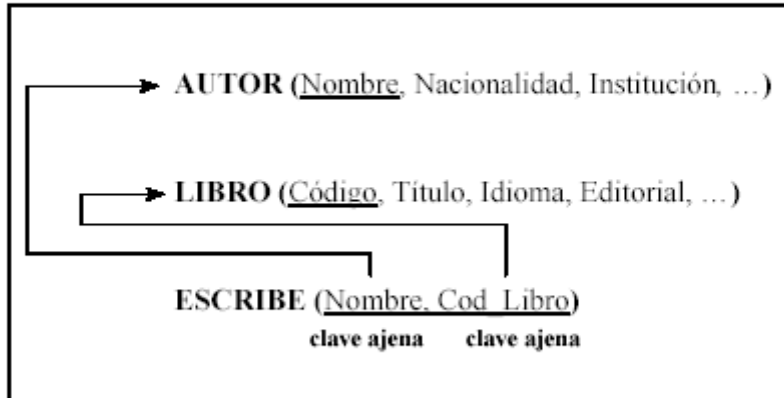
Así, en la tabla personal, las claves candidatas serán DNI y NOMBRE.

En ambas el atributo cuyos valores ocupan menos memoria es el DNI, además el acceso a él será más rápido, aparte de que no se necesita ningún tipo de algoritmo para codificarlo.

Se denomina CLAVE AJENA a aquel atributo o conjunto de atributos que en la tabla donde se encuentran no son clave, y sus valores se corresponden con la clave principal de la misma u otra tabla. El dominio de la clave ajena y de la clave principal de la tabla a la que hace referencia se tiene que corresponder, es decir, ser compatible. Por la tanto el atributo DNI por se clave principal de "personal" y estar en una columna de la relación "coches-personal" es clave ajena de ésta última tabla.

Las claves ajenas son un concepto muy importante, puesto que la integridad de la base de datos se mantiene o se elimina en una gran parte mediante las transacciones que se realizan sobre dichas claves.

Ejemplo de claves primarias y ajenas



Los atributos principales (forman la clave primaria) se subrayan

Los mecanismos que proporciona el modelo relacional para recoger restricciones de usuario son las siguientes:

- a) Restricción de clave primaria (PRIMARY KEY) permite declarar un atributo o conjunto de atributos como la clave primaria de una relación (identifica unívocamente cada tupla de la relación). En el ejemplo de la relación ALUMNO, la clave primaria sería Cod_Matrícula.
- b) La restricción de unicidad (UNIQUE) nos permite definir claves alternativas (los valores de uno o varios atributos no pueden repetirse en diferentes tuplas de una relación)
- c) La restricción de obligatoriedad (NOT NULL) permite declarar si uno o varios atributos de una relación deben tomar siempre un valor, es decir, no pueden tomar valores nulos.
- d) La restricción de clave ajena (FOREIGN KEY), también denominada integridad referencial, se usa para, mediante claves ajenas (conjunto de atributos en una relación que es una clave primaria en otra o la misma relación) enlazar relaciones de la base de datos. La integridad referencial nos indica que los valores de la clave ajena en la relación hijo deben corresponderse con los valores de la clave primaria en la relación padre o bien ser nulos si se admiten nulos. Los atributos que son clave ajena en una relación no necesitan tener los mismos nombres que los atributos de la clave primaria con la cual ellos se corresponden

<i>Nombre_e</i>	<i>Dirección</i>	<i>País</i>	<i>Ciudad</i>
Universal Books	Brown Sq. 23	EEUU	Los Angeles
Rama	Canillas, 144	España	Madrid
Mc Graw-Hill	Basauri 17	España	Madrid
Paraninfo	Virtudes 7	España	Madrid

EDITORIAL

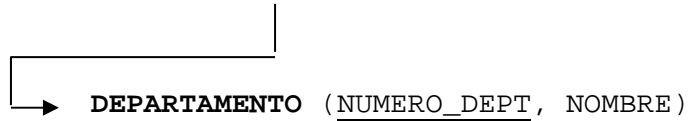
LIBRO

<i>Código</i>	<i>Título</i>	<i>...</i>	<i>Editorial</i>
00345D7	Int. Artificial		Paraninfo
1022305	Concep. y Dis.		Rama
4939H2	Turbo C++		Mc Graw-Hill
0045307	Virus Informát.		Nulo
01123J3	Sist. Informac.		Rama

*Ejemplo de
Integridad
Referencial*

Otro ejemplo:

EMPLEADO (NOMBRE, DEPARTAMENTO, SALARIO, FECH_NAC, EXT_TF)



Las claves primarias de la relación hija EMPLEADO y de la relación padre DEPARTAMENTO son "Nombre" y "Número_dpto", respectivamente. El atributo "Departamento" de la relación EMPLEADO es una clave ajena que referencia la relación DEPARTAMENTO, es decir, los valores del atributo "Departamento" deben corresponderse con los valores del atributo clave primaria "Número_dpto" de la relación DEPARTAMENTO. Algunas tuplas de estas relaciones se muestran a continuación:

NOMBRE	DEPARTAM.	SALARIO	FECH_NAC	EXT_TF
Pablo Montero	14	220.000	10-11-67	6543
Bea Cristobal	13	300.000	20-9-68	6577
J. Luis Martín	11	150.000	25-6-77	6433
Plinky Agudo	13	350.000	3-7-80	6422
Kris Iron	14	400.000	14-3-75	6321
Vanesa Paradisse	11	200.000	12-3-70	6323

"EMPLEADO"

NUMERO_DEPARTAMENTO	NOMBRE
11	Contabilidad
13	Marketing
14	Informática

"DEPARTAMENTO"

Además, es necesario determinar las consecuencias de ciertas operaciones (borrado y modificación) realizadas sobre tuplas de la relación referenciada, es decir, las opciones de borrado y modificación sobre claves ajenas. Las posibilidades para una operación de actualización (borrado o modificación) son:

- Borrado / modificación en cascada (CASCADE): el borrado (o modificación) de una tupla en la relación padre ocasiona un borrado (o modificación) de todas las tuplas relacionadas en la relación hija (tuplas cuya clave ajena coincida con el valor de la clave primaria de la tupla eliminada o modificada en la relación padre). Así, en el ejemplo anterior de EMPLEADOS y DEPARTAMENTOS, el borrado de un departamento supone el borrado de todos los empleados que trabajan en él.
- Borrado / modificación restringido (RESTRICT): en este caso si existen tuplas en la relación hija relacionadas con la tupla de la relación padre sobre la que se realiza la operación, entonces no se permitirá llevar a cabo dicha operación. En el ejemplo, no se podría borrar un departamento que tenga empleados que trabajen en él.
- Borrado / modificación con puesta a nulos (SET NULL): esta posibilidad nos permite poner el valor de la clave ajena referenciada a NULL cuando se produce el borrado o modificación de una tupla en la relación padre. En el ejemplo, el borrado de un departamento implicaría que todos los empleados que trabajan en él pasarán a contener NULL en el atributo DEPARTAMENTO. Evidentemente, esta opción está permitida siempre y cuando la clave ajena admita nulas.
- Borrado / modificación con puesta a un valor por defecto (SET DEFAULT): su funcionamiento es similar al caso anterior, con la excepción de que el valor al que se ponen las claves ajenas referenciadas es un valor por defecto que se habrá especificado en la definición de la tabla correspondiente. En este caso, en el ejemplo anterior, cuando se borra un determinado departamento es posible asignar sus empleados a un departamento ficticio (que debe encontrarse evidentemente presente en la relación DEPARTAMENTO)

Las opciones de borrado y modificación pueden ser distintas para una determinada clave ajena de una relación; por ejemplo, es posible definir el borrado en cascada y la modificación restringida.

e) Restricciones de Verificación (CHECK): En algunos casos puede ocurrir que sea necesario especificar una condición que deben cumplir los valores de determinados atributos de una relación de la BdD aparte de las restricciones ya vistas de clave primaria, unicidad, obligatoriedad y clave ajena. Para ello se definen las restricciones de verificación que, al igual que las ya estudiadas, siempre llevan implícitas un rechazo en caso de que no se cumpla la condición especificada y que también se comprueban ante una inserción, borrado o modificación. Como las restricciones de verificación siempre van ligadas a un único elemento de la BdD (generalmente una relación) no es necesario que tengan un nombre. Por ejemplo, para la relación EMPLEADO podría definirse una restricción sobre el atributo "Salario" que estableciera que "el rango del salario del empleado puede oscilar entre las 100.000 y las 500.000 pesetas. Así, si se va a insertar un empleado con un sueldo inferior a 100.000, la operación se rechazaría.

f) Aserciones (ASSERTION): Otro tipo de restricción, que generaliza al anterior, lo forman las aserciones en las que la condición se establece sobre elementos de distintas relaciones (por ello debe tener un nombre que la identifique). Su funcionamiento es idéntico al de las restricciones de verificación. La única diferencia consiste en que las restricciones de verificación o CHECK tienen como ámbito una única relación. En el ejemplo, se podría especificar una aserción para la restricción que establezca que "no hay ningún empleado que trabaje en el departamento de contabilidad que gane más de 200.000 pesetas".

En las restricciones que involucran algún tipo de subconsulta en la condición (por ejemplo, si se usa alguna función de agregación sobre las tuplas de una relación) puede ocurrir que puedan implementarse como un CHECK. Sin embargo, es necesario estudiar que la restricción no pueda ser violada en ningún caso por cualquiera de las operaciones que puedan realizarse sobre el esquema. Por ello, a veces es necesario implementarlas como una aserción.

g) Disparadores (TRIGGER): Por último, a veces puede interesar especificar una acción distinta del rechazo cuando no se cumple una determinada restricción. En este caso, se recurre al uso de los triggers que nos permiten además de indicar una condición, especificar la acción que queremos se lleve a cabo si la condición se hace verdadera. Los disparadores pueden interpretarse como reglas de tipo evento-condición-acción (ECA) que pueden interpretarse como reglas que especifican que cuando se produce un evento, si se cumple una condición, entonces se realiza una determinada acción. Los triggers no están soportados en ciertas versiones de SQL, sin embargo, existen productos relacionales que los soportan aunque

con diferencias en el modo de funcionamiento. En el ejemplo, se podría definir un trigger que "informará al administrador de la BdD cuando haya un empleado que gane más de 450.000 pts y que trabaje en el departamento de Marketing".

⇒ NOTACIÓN

Un esquema relacional se representa mediante un grafo, conocido como grafo relacional. Se trata de un grafo dirigido cuyos nodos son las relaciones de la BdD y los arcos representan las restricciones de clave ajena, y en el que aparecerán además de las distintas relaciones con sus atributos y las restricciones de clave ajena las restricciones de clave primaria, unicidad y obligatoriedad. Las convenciones empleadas para la representación de este grafo son:

El nombre de las tablas estará representado en mayúsculas (igual que los atributos) y en negrita. Primero aparecer el nombre de la relación y después los atributos entre paréntesis.

- Las claves primarias aparecen subrayadas
- Las claves alternativas aparecen en negrita
- Las claves ajenas están representadas en letra cursiva y referencian a la relación en la que son clave primaria mediante una flecha.
- Los atributos que pueden tomar valores nulos aparecen con asterisco.

- Las opciones de integridad referencial son:

- B:C, Borrado en Cascada
- B:N, Borrado con puesta a Nulos
- B:D, Borrado con puesta a valor por defecto
- B:R, Borrado restringido
- M:C, Modificación en cascada
- M:N, Modificación con puesta a Nulos
- M:D, Modificación con puesta a valor por defecto
- M:R, Modificación Restringida.

3.- PASO DEL MODELO ENTIDAD/RELACIÓN AL MODELO RELACIONAL

INTERRELACIONES 1:1 Entre dos entidades se asigna la la clave de la entidad más importante para el sistema a la entidad menos importante.

INTERRELACIONES 1:N Se agrega la clave de la entidad 1 a la entidad de lado N.

INTERRELACCIONES N:N Dan lugar a una entidad renacida cuya clave estará compuesta por la unión de las claves que la producen.

4.- EJERCICIOS

Pasar al modelo relacional los ejercicios del capítulo entidad/relación.