

TEMA 4: ELABORACIÓN DEL DISEÑO LÓGICO. MODELO RELACIONAL.

1ª PARTE

INTRODUCCIÓN Y ESTRUCTURA DE LOS DATOS

OBJETIVOS:

- Conocer los fundamentos del modelo de datos relacional.
- Comprender las ventajas del modelo relacional que derivan del alto grado de independencia de los datos que proporciona, y de la simplicidad y la uniformidad del modelo.
- Conocer las operaciones del álgebra relacional.
- Saber utilizar las operaciones del álgebra relacional para consultar una base de datos.



INTRODUCCIÓN Y ESTRUCTURA DE LOS DATOS

Introducción

Esta unidad didáctica está dedicada al estudio del modelo de datos relacional y del álgebra relacional.

El concepto de modelo de datos se ha presentado en la primera unidad. En ésta se profundiza en un modelo de datos concreto: el modelo relacional, que actualmente tiene una gran relevancia. Sus conceptos fundamentales están bien asentados y, además, los sistemas de gestión de bases de datos relacionales son los más extendidos en su utilización práctica.

El estudio del modelo relacional sirve, además, de base para los contenidos de otra unidad, dedicada al lenguaje SQL. Este lenguaje permite definir y manipular bases de datos relacionales. Los fundamentos del modelo relacional resultan imprescindibles para conseguir un buen dominio del SQL.

En esta unidad se analizan también las operaciones del álgebra relacional, que sirven para hacer consultas a una base de datos. Es preciso conocer estas operaciones porque nos permiten saber qué servicios de consulta debe proporcionar un lenguaje relacional. Otra aportación del álgebra relacional es que facilita la comprensión de algunas de las construcciones del lenguaje SQL que se estudiarán en otra unidad didáctica de este curso. Además, constituye la base para el estudio del tratamiento de las consultas que efectúan los SGBD internamente (especialmente en lo que respecta a la optimización de consultas).

Primera aproximación

- Este modelo está basado en la teoría matemática de las relaciones introducida por Codd a finales de los años sesenta.
- Esta teoría supuso un importante paso en la investigación de los SGBD, suministrando un sólido fundamento de desarrollo.
- Codd propone un modelo basado en esta teoría donde los datos se estructuran lógicamente en relaciones, que al final se constituyen en tablas.
- Una relación, en terminología relacional, es un conjunto de filas (tuplas) con unas determinadas características.

- Con respecto a la parte dinámica del modelo, se propone un conjunto de operadores que se aplican a las relaciones. Estos operadores conforman lo que se denomina álgebra relacional.
- La teoría de la normalización cuyas tres primeras formas normales fueron introducidas por Codd, proporciona una estructura más regular en la representación de relaciones. (se verá en la unidad nº6)

1. Introducción al modelo relacional

El modelo relacional es un modelo de datos y, como tal, tiene en cuenta los tres aspectos siguientes de los datos:

- 1) La estructura, que debe permitir representar la información que nos interesa del mundo real.
- 2) La manipulación, a la que da apoyo mediante las operaciones de actualización y consulta de los datos.
- 3) La integridad, que es facilitada mediante el establecimiento de reglas de integridad; es decir, condiciones que los datos deben cumplir.

Un sistema de gestión de bases de datos relacional (SGBDR) da apoyo a la definición de datos mediante la estructura de los datos del modelo relacional, así como a la manipulación de estos datos con las operaciones del modelo; además, asegura que se satisfacen las reglas de integridad que el modelo relacional establece.

Los principios del modelo de datos relacional fueron establecidos por E.F. Codd en los años 1969 y 1970. De todos modos, hasta la década de los ochenta no se empezaron a comercializar los primeros SGBD relacionales con rendimientos aceptables. Cabe señalar que los SGBD relacionales que se comercializan actualmente todavía no soportan todo lo que establece la teoría relacional hasta el último detalle.

La aparición del modelo relacional representa un verdadero hito en el desarrollo de bases de datos, ya que marca tres etapas.

- Etapa prerrelacional (. . . – 1978): es la primera generación de bases de datos, en la que los SGBD se soportan en los modelos Codasyl y Jerárquico.
- Etapa relacional (1979 – 1990): es la segunda generación de bases de datos, en la que los sistemas relacionales se van aproximando a su madurez y los productos basados en este modelo van desplazando poco a poco a los sistemas de primera generación hasta conseguir una mayor cuota en el mercado de las bases de datos.
- Etapa postrelacional (1991 – . . .): es la tercera generación de bases de datos, en la que aparecen otros modelos de datos, en especial orientados al objeto, que están en estos momentos intentando abrirse un hueco en el mercado de las bases de datos. Sin embargo, casi nadie se atreve a vaticinar, al menos a corto plazo, una sustitución de los sistemas relacionales por sistemas basados en otros modelos.

El principal objetivo del modelo de datos relacional es facilitar que la base de datos sea percibida o vista por el usuario como una estructura lógica que consiste en un conjunto de relaciones y no como una estructura física de implementación. Esto ayuda a conseguir un alto grado de independencia de los datos.

Un objetivo adicional del modelo es conseguir que esta estructura lógica con la que se percibe la base de datos sea simple y uniforme. Con el fin de proporcionar simplicidad y uniformidad, toda la información se representa de una única manera: mediante valores explícitos que contienen las relaciones (no se utilizan conceptos como por ejemplo apuntadores entre las relaciones). Con el mismo propósito, todos los valores de datos se consideran atómicos; es decir, no es posible descomponerlos.

Hay que precisar que un SGBD relacional, en el nivel físico, puede emplear cualquier estructura de datos para implementar la estructura lógica formada por las relaciones. En particular, a nivel físico, el sistema puede utilizar apuntadores, índices, etc. Sin embargo, esta implementación física queda oculta al usuario.

En los siguientes apartados estudiaremos la estructura de los datos, las operaciones y las reglas de integridad del modelo relacional.

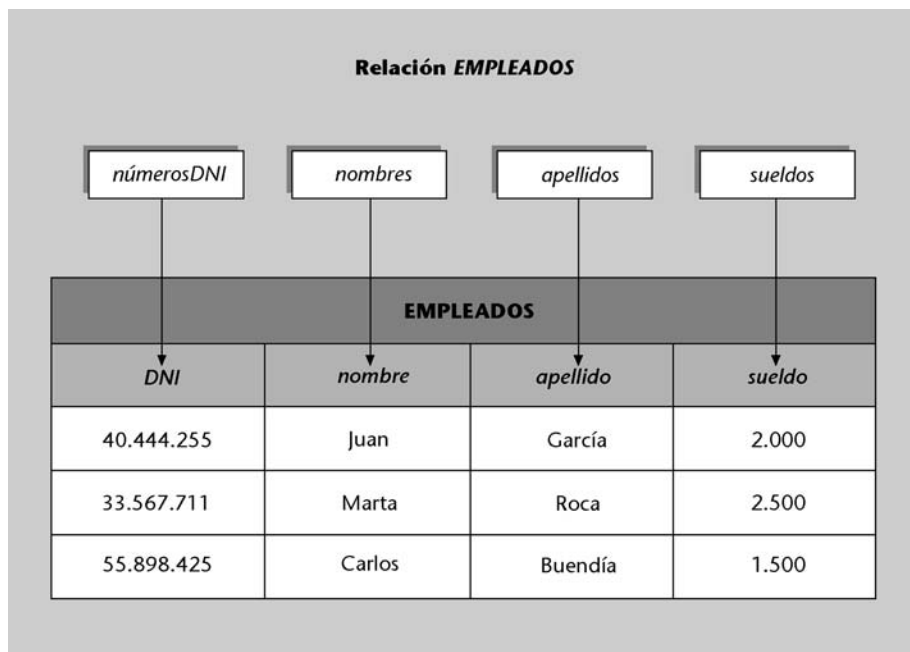
2.- ESTRUCTURA DE LOS DATOS

El modelo relacional proporciona una estructura de los datos que consiste en un conjunto de relaciones con objeto de representar la información que nos interesa del mundo real.

La estructura de los datos del modelo relacional se basa, pues, en el **concepto de relación**.

2.1. Visión informal de una relación

En primer lugar, presentaremos el concepto de relación de manera informal. Se puede obtener una buena idea intuitiva de lo que es una relación si la visualizamos como una tabla o un fichero. En la figura 1 se muestra la visualización tabular de una relación que contiene datos de empleados. Cada fila de la tabla contiene una colección de valores de datos relacionados entre sí; en nuestro ejemplo, son los datos correspondientes a un mismo empleado. La tabla tiene un nombre (EMPLEADOS) y también tiene un nombre cada una de sus columnas (DNI, nombre, apellido y sueldo). El nombre de la tabla y los de las columnas ayudan a entender el significado de los valores que contiene la tabla. Cada columna contiene valores de un cierto dominio; por ejemplo, la columna DNI contiene valores del dominio númerosDNI.



Si definimos las relaciones de forma más precisa, nos daremos cuenta de que presentan algunas características importantes que, en la visión superficial que hemos presentado, quedan ocultas. Estas características son las que motivan que el concepto de relación sea

totalmente diferente del de fichero, a pesar de que, a primera vista, relaciones y ficheros puedan parecer similares.

2.2. Visión formal de una relación

A continuación definimos formalmente las relaciones y otros conceptos que están vinculados a ellas, como por ejemplo dominio, esquema de relación, etc.

Un dominio D es un conjunto de valores atómicos. Por lo que respecta al modelo relacional, atómico significa indivisible; es decir, que por muy complejo o largo que sea un valor atómico, no tiene una estructuración interna para un SGBD relacional.

Los dominios pueden ser de dos tipos:

- 1) Dominios predefinidos, que corresponde a los tipos de datos que normalmente proporcionan los lenguajes de bases de datos, como por ejemplo los enteros, las cadenas de caracteres, los reales, etc.
- 2) Dominios definidos por el usuario, que pueden ser más específicos. Toda definición de un dominio debe constar, como mínimo, del nombre del dominio y de la descripción de los valores que forman parte de éste. Por ejemplo, el usuario puede definir un dominio para las edades de los empleados que se denomine dom_edad y que contenga los valores enteros que están entre 16 y 65

Intensión y Extensión:

Una **relación** se compone del **esquema** (o **intensión** de la relación) y de la **extensión**.

- El esquema de relación representa la parte definitoria y estática de la relación y se denomina **intensión**.

Se corresponde con el concepto de **entidad** en el modelo Entidad/Relación.

- El estado de relación representa el conjunto de tuplas que, en un instante dado, están presentes en el esquema de relación, y se denomina **extensión**. Se corresponde con el concepto de **ejemplar** en el modelo Entidad/Relación.

- El esquema de la relación consiste en un nombre de relación R y un conjunto de atributos $\{A_1, A_2, \dots, A_n\}$.

Empleados			
DNI	nombre	apellido	sueldo
40.444.255	Juan	García	2.000
33.567.711	Marta	Roca	2.500
55.898.425	Carlos	Buendía	1.500

Esquema

Extensión

Si tomamos como ejemplo la figura 1, el nombre de la relación es EMPLEADOS y el conjunto de atributos es $\{DNI, nombre, apellido, sueldo\}$.

Tomaremos la convención de denotar el esquema de la relación de la forma siguiente: $R(A_1, A_2, \dots, A_n)$, donde R es el nombre de la relación y A_1, A_2, \dots, A_n es una ordenación cualquiera de los atributos que pertenecen al conjunto $\{A_1, A_2, \dots, A_n\}$.

El esquema de la relación de la figura 1 se podría denotar, por ejemplo, como EMPLEADOS(DNI, nombre, apellido, sueldo), o también, EMPLEADOS(nombre, apellido, DNI, sueldo), porque cualquier ordenación de sus atributos se considera válida para denotar el esquema de una relación.

- Un atributo A_i es el nombre del papel que ejerce un dominio D en un esquema de relación.
 D es el dominio de A_i y se denota como dominio (A_i).

Dominio del atributo DNI:

Según la figura 1, el atributo DNI corresponde al papel que ejerce el dominio númerosDNI en el esquema de la relación EMPLEADOS y, entonces, dominio(DNI) = númerosDNI.

Conviene observar que cada atributo es único en un esquema de relación, porque no tiene sentido que un mismo dominio ejerza dos veces el mismo papel en un mismo esquema. Por consiguiente, no puede ocurrir que en un esquema de relación haya dos atributos con el mismo nombre. En cambio, sí que se puede repetir un nombre de atributo en relaciones diferentes. Los dominios de los atributos, por el contrario, no deben ser necesariamente todos diferentes en una relación.

Ejemplo de atributos diferentes con el mismo dominio:

Si tomamos como ejemplo el esquema de relación PERSONAS(DNI, nombre, apellido, telcasa, teltrabajo), los atributos telcasa y teltrabajo pueden tener el mismo dominio: $\text{dominio}(\text{telcasa}) = \text{teléfono}$ y $\text{dominio}(\text{teltrabajo}) = \text{teléfono}$.

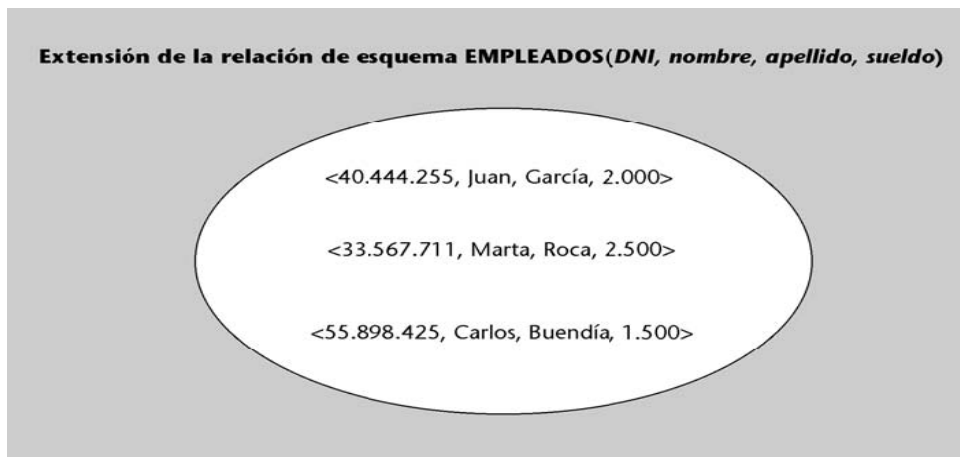
En este caso, el dominio teléfono ejerce dos papeles diferentes en el esquema de relación: el de indicar el teléfono particular de una persona y el de indicar el del trabajo.

- La extensión de la relación de esquema $R(A_1, A_2, \dots, A_n)$ es un conjunto de tuplas t_i ($i = 1, 2, \dots, m$), donde cada tupla t_i es, a su vez un conjunto de pares $t_i = \{ \langle A_1:v_{i1} \rangle, \langle A_2:v_{i2} \rangle \dots \langle A_n:v_{in} \rangle \}$ y, para cada par $\langle A_j:v_{ij} \rangle$, se cumple que v_{ij} es un valor de $\text{dominio}(A_j)$, o bien un valor especial que denominaremos nulo.

Para simplificar, tomaremos la convención de referirnos a una tupla $t_i = \{ \langle A_1:v_{i1} \rangle, \langle A_2:v_{i2} \rangle, \dots, \langle A_n:v_{in} \rangle \}$ que pertenece a la extensión del esquema denotado como $R(A_1, A_2, \dots, A_n)$, de la forma siguiente: $t_i = \langle v_{i1}, v_{i2}, \dots, v_{in} \rangle$.

Si denotamos el esquema de la relación representada en la figura 1 como EMPLEADOS(DNI, nombre, apellido, sueldo), el conjunto de tuplas de su extensión será el de la figura siguiente:

Figura 3



Si en una tupla $t_i = \langle v_{i1}, v_{i2}, \dots, v_{in} \rangle$, el valor v_{ij} es un valor nulo, entonces el valor del atributo A_j es desconocido para la tupla t_i de la relación, o bien no es aplicable a esta tupla.

Ejemplo de valor nulo:

Podríamos tener un atributo telcasa en la relación EMPLEADOS y se podría dar el caso de que un empleado no tuviese teléfono en su casa, o bien que lo tuviese, pero no se conociese su número. En las dos situaciones, el valor del atributo telcasa para la tupla correspondiente al empleado sería el valor nulo.

- El **grado** de una relación es el número de atributos que pertenecen a su esquema.

Grado de la relación EMPLEADOS

El grado de la relación de esquema EMPLEADOS(DNI, nombre, apellido, sueldo), es **4**.

- La **cardinalidad** de una relación es el número de tuplas que pertenecen a su extensión.

Cardinalidad de la relación EMPLEADOS

Observando la figura 3 se deduce que la cardinalidad de la relación EMPLEADOS es **3**.

Resumiendo:

- **Nombre:** nombre de la relación (puede haber relaciones, como aquellas que se obtienen como resultado de una consulta, que no tienen nombre).
- **Atributos:** conjunto de columnas que representan las propiedades de la tabla y que también están caracterizadas por su nombre.
- **Tuplas:** conjunto de filas que contienen los valores que toma cada uno de los atributos por cada elemento de la relación.
- **Dominios:** valores que pueden tomar los atributos.
- **Grado:** número de atributos.
- **Cardinalidad:** número de tuplas.

2.3. Diferencias entre relaciones y ficheros

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

A primera vista, relaciones y ficheros resultan similares. Los registros y los campos que forman los ficheros se parecen a las tuplas y a los atributos de las relaciones, respectivamente. A pesar de esta similitud superficial, la visión formal de relación que hemos presentado establece algunas características de las relaciones que las hacen diferentes de los ficheros clásicos. A continuación describimos estas características:

1) **Atomicidad de los valores de los atributos:** los valores de los atributos de una relación deben ser atómicos; es decir, no deben tener estructura interna. Esta característica proviene del hecho de que los atributos siempre deben tomar un valor de su dominio o bien un valor nulo, y de que se ha establecido que los valores de los dominios deben ser atómicos en el modelo relacional.

El objetivo de la atomicidad de los valores es dar simplicidad y uniformidad al modelo relacional.

2) **No-repetición de las tuplas:** en un fichero clásico puede ocurrir que dos de los registros sean exactamente iguales; es decir, que contengan los mismos datos. En el caso del modelo relacional, en cambio, no es posible que una relación contenga tuplas repetidas. Esta característica se deduce de la misma definición de la extensión de una relación. La extensión es un conjunto de tuplas y, en un conjunto, no puede haber elementos repetidos.

3) **No-ordenación de las tuplas:** de la definición de la extensión de una relación como un conjunto de tuplas se deduce también que estas tuplas no estarán ordenadas, teniendo en cuenta que no es posible que haya una ordenación entre los elementos de un conjunto.

La finalidad de esta característica es conseguir que, mediante el modelo relacional, se puedan representar los hechos en un nivel abstracto que sea independiente de su estructura física de implementación. Más concretamente, aunque los SGBD relacionales deban proporcionar una implementación física que almacenará las tuplas de las relaciones en un orden concreto, esta ordenación no es visible si nos situamos en el nivel conceptual.

4) **No-ordenación de los atributos:** el esquema de una relación consta de un nombre de relación R y un conjunto de atributos $\{A_1, A_2, \dots, A_n\}$. Así pues, no hay un orden entre los atributos de un esquema de relación, teniendo en cuenta que estos atributos forman un conjunto.

Por ejemplo, ...

... si se almacena información sobre los empleados de una empresa, es preciso tener la posibilidad de distinguir qué datos corresponden a cada uno de los diferentes empleados.

Como en el caso anterior, el objetivo de esta característica es representar los hechos en un nivel abstracto, independientemente de su implementación física.

Ejemplo de no-ordenación de los atributos

El esquema de relación EMPLEADOS(DNI, nombre, apellido, sueldo) denota el mismo esquema de relación que EMPLEADOS(nombre, apellido, DNI, sueldo).

2.4. Clave candidata, clave primaria y clave alternativa de las relaciones

Toda la información que contiene una base de datos debe poderse identificar de alguna forma. En el caso particular de las bases de datos que siguen el modelo relacional, para identificar los datos que la base de datos contiene, se pueden utilizar las claves candidatas de las relaciones. A continuación definimos qué se entiende por clave candidata, clave primaria y clave alternativa de una relación. Para hacerlo, será necesario definir el concepto de **superclave**.

Observad que...

... toda relación tiene, por lo menos, una superclave, que es la formada por todos los atributos de su esquema. Esto se debe a la propiedad que cumple toda relación de no tener tuplas repetidas. En el ejemplo de EMPLEADOS(DNI, NSS, nombre, apellido, teléfono) esta superclave sería: {DNI, NSS, nombre, apellido, teléfono}.

- Una **superclave** de una relación de esquema $R(A_1, A_2, \dots, A_n)$ es un subconjunto de los atributos del esquema tal que no puede haber dos tuplas en la extensión de la relación que tengan la misma combinación de valores para los atributos del subconjunto.

Una superclave, por lo tanto, nos permite identificar todas las tuplas que contiene la relación.

Algunas superclaves de la relación EMPLEADOS:

En la relación de esquema EMPLEADOS(DNI, NSS, nombre, apellido, teléfono), algunas de las superclaves de la relación serían los siguientes subconjuntos de atributos: {DNI, NSS, nombre, apellido, teléfono}, {DNI, apellido}, {DNI} y {NSS}.

- Una **clave candidata** (C) de una relación es una **superclave** de la relación, que cumple que ningún subconjunto propio de C es superclave.

Es decir, C cumple que la eliminación de cualquiera de sus atributos da un conjunto de atributos que no es superclave de la relación. Intuitivamente, una clave candidata permite identificar cualquier tupla de una relación, de manera que no sobre ningún atributo para hacer la identificación.

Claves candidatas de EMPLEADOS :

En la relación de esquema EMPLEADOS(DNI, NSS, nombre, apellido, teléfono), sólo hay dos claves candidatas: {DNI} y {NSS}.

- Habitualmente, una de las claves candidatas de una relación se designa **clave primaria** de la relación. La **clave primaria** (CP) es la clave candidata cuyos valores se utilizarán para identificar las tuplas de la relación.

El diseñador de la base de datos es quien elige la clave primaria de entre las claves candidatas.

Las claves candidatas no elegidas como primaria se denominan **claves alternativas**.

Utilizaremos la convención de subrayar los atributos que forman parte de la clave primaria en el esquema de la relación. Así pues, R(A1, A2, ..., Ai, ..., An) indica que los atributos A1, A2, ..., Ai forman la clave primaria de R.

Elección de la clave primaria de EMPLEADOS

En la relación de esquema EMPLEADOS(DNI, NSS, nombre, apellido, teléfono), donde hay dos claves candidatas, {DNI} y {NSS}, se puede elegir como clave primaria {DNI}. Lo indicaremos subrayando el atributo DNI en el esquema de la relación EMPLEADOS(DNI, NSS, nombre, apellido, teléfono). En este caso, la clave {NSS} será una clave alternativa de EMPLEADOS.

Notad que, ...

... puesto que toda relación tiene por lo menos una superclave, podemos garantizar que toda relación tiene como mínimo una clave candidata.

Es posible que una clave candidata o una clave primaria consten de más de un atributo.

Clave primaria de la relación DESPACHOS:

En la relación de esquema DESPACHOS(edificio, número, superficie), la clave primaria está formada por los atributos edificio y número. En este caso, podrá ocurrir que dos despachos diferentes estén en el mismo edificio, o bien que tengan el mismo número, pero nunca pasará que tengan la misma combinación de valores para edificio y número.

Relación con una clave candidata

Si una relación sólo tiene una clave candidata, entonces esta clave candidata debe ser también su clave primaria. Ya que todas las relaciones tienen como mínimo una clave candidata, podemos garantizar que, para toda relación, será posible designar una clave primaria.

2.5. Claves foráneas de las relaciones

Hasta ahora hemos estudiado las relaciones de forma individual, pero debemos tener en cuenta que una base de datos relacional normalmente contiene más de una relación, para poder representar distintos tipos de hechos que suceden en el mundo real. Por ejemplo, podríamos tener una pequeña base de datos que contuviese dos relaciones: una denominada EMPLEADOS, que almacenaría datos de los empleados de una empresa, y otra con el nombre DESPACHOS, que almacenaría los datos de los despachos que tiene la empresa.

Debemos considerar también que entre los distintos hechos que se dan en el mundo real pueden existir lazos o vínculos. Por ejemplo, los empleados que trabajan para una empresa pueden estar vinculados con los despachos de la empresa, porque a cada empleado se le asigna un despacho concreto para trabajar.

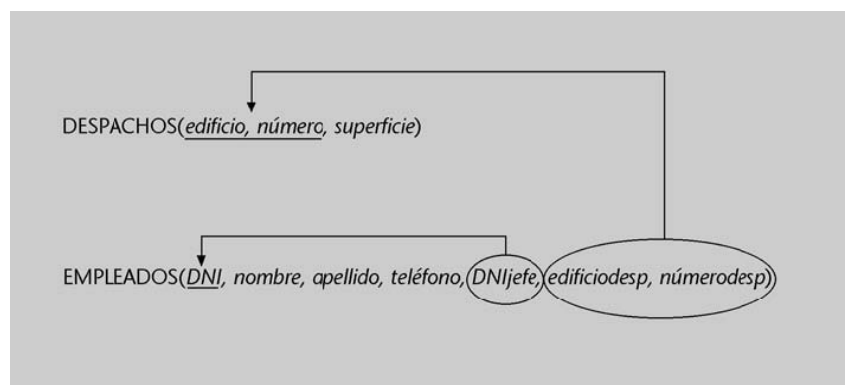
En el modelo relacional, para reflejar este tipo de vínculos, tenemos la posibilidad de expresar conexiones entre las distintas tuplas de las relaciones. Por ejemplo, en la base de datos anterior, que tiene las relaciones EMPLEADOS y DESPACHOS, puede ser necesario conectar tuplas de EMPLEADOS con tuplas de DESPACHOS para indicar qué despacho tiene asignado cada empleado.

En ocasiones, incluso puede ser necesario reflejar lazos entre tuplas que pertenecen a una misma relación. Por ejemplo, en la misma base de datos anterior puede ser necesario conectar determinadas tuplas de EMPLEADOS con otras tuplas de EMPLEADOS para indicar, para cada empleado, quién actúa como su jefe.

El mecanismo que proporcionan las bases de datos relacionales para conectar tuplas son las **claves foráneas** de las relaciones. Las **claves foráneas** permiten establecer conexiones entre las tuplas de las relaciones. Para hacer la conexión, una clave foránea tiene el conjunto de atributos de una relación que referencian la clave primaria de otra relación (o incluso de la misma relación).

Claves foráneas de la relación EMPLEADOS

En la figura siguiente, la relación EMPLEADOS(DNI, nombre, apellido, teléfono, DNIjefe, edificiodesp, númerodesp), tiene una clave foránea formada por los atributos edificiodesp y númerodesp que se refiere a la clave primaria de la relación DESPACHOS(edificio, número, superficie). Esta clave foránea indica, para cada empleado, el despacho donde trabaja. Además, el atributo DNIjefe es otra clave foránea que referencia la clave primaria de la misma relación EMPLEADOS, e indica, para cada empleado, quien es su jefe.



Las claves foráneas tienen por objetivo establecer una conexión con la clave primaria que referencian. Por lo tanto, los valores de una clave foránea deben estar presentes en la clave primaria correspondiente, o bien deben ser valores nulos. En caso contrario, la clave foránea representaría una referencia o conexión incorrecta.

Ejemplo:

En la relación de esquema EMPLEADOS(DNI, nombre, apellido, DNIjefe, edificiodesp, númeroesp), la clave foránea {edificiodesp, númeroesp} referencia la relación DESPACHOS(edificio, número, superficie). De este modo, se cumple que todos los valores que no son nulos de los atributos edificiodesp y númeroesp son valores que existen para los atributos edificio y número de DESPACHOS, tal y como se puede ver a continuación:

• Relación DESPACHOS:

<i>edificio</i>	<i>número</i>	<i>superficie</i>
Marina	120	10
Marina	122	15
Marina	230	20
Diagonal	120	10

• Relación EMPLEADOS:

<i>DNI</i>	<i>nombre</i>	<i>apellido</i>	<i>DNIjefe</i>	<i>edificiodesp</i>	<i>númeroesp</i>
40.444.255	Juan	García	NULO	Marina	120
33.567.711	Marta	Roca	40.444.255	Marina	120
55.898.425	Carlos	Buendía	40.444.255	Diagonal	120
77.232.144	Elena	Pla	40.444.255	NULO	NULO

Supongamos que hubiese un empleado con los valores <55.555.555, María, Casagran, NULO, París, 400>. Puesto que no hay ningún despacho con los valores París y 400 para edificio y número, la tupla de este empleado hace una referencia incorrecta; es decir, indica un despacho para el empleado que, de hecho, no existe.

Es preciso señalar que en la relación EMPLEADOS hay otra clave foránea, {DNIjefe}, que referencia la misma relación EMPLEADOS, y entonces se cumple que todos los valores que no son nulos del atributo DNIjefe son valores que existen para el atributo DNI de la misma relación EMPLEADOS.

A continuación estableceremos de forma más precisa qué se entiende por clave foránea.

- Una clave foránea de una relación R es un subconjunto de atributos del esquema de la relación, que denominamos CF y que cumple las siguientes condiciones:

1) Existe una relación S (S no debe ser necesariamente diferente de R) que tiene por clave primaria CP.

2) Se cumple que, para toda tupla t de la extensión de R , los valores para CF de t son valores nulos o bien valores que coinciden con los valores para CP de alguna tupla s de S .

Y entonces, se dice que la clave foránea CF referencia la clave primaria CP de la relación S , y también que la clave foránea CF referencia la relación S .

De la noción que hemos dado de clave foránea se pueden extraer varias consecuencias:

1) Si una clave foránea CF referencia una clave primaria CP , el número de atributos de CF y de CP debe coincidir.

Ejemplo de coincidencia del número de atributos de CF y CP

En el ejemplo anterior, tanto la clave foránea {edificiodesp, númeroesp} como la clave primaria que referencia {edificio, número} tienen dos atributos. Si no sucediese así, no sería posible que los valores de CF existieran en CP .

2) Por el mismo motivo, se puede establecer una correspondencia (en concreto, una biyección) entre los atributos de la clave foránea y los atributos de la clave primaria que referencia.

Ejemplo de correspondencia entre los atributos de CF y los de CP

En el ejemplo anterior, a edificiodesp le corresponde el atributo edificio, y a númeroesp le corresponde el atributo número.

3) También se deduce de la noción de clave foránea que los dominios de sus atributos deben coincidir con los dominios de los atributos correspondientes a la clave primaria que referencia. Esta coincidencia de dominios hace que sea posible que los valores de la clave foránea coincidan con valores de la clave primaria referenciada.

Ejemplo de coincidencia de los dominios

En el ejemplo anterior, se debe cumplir que $\text{dominio}(\text{edificiodesp}) = \text{dominio}(\text{edificio})$ y también que $\text{dominio}(\text{númeroesp}) = \text{dominio}(\text{número})$.

Observad que, de hecho, esta condición se podría relajar, y se podría permitir que los dominios no fuesen exactamente iguales, sino que sólo fuesen, y de alguna forma que convendría precisar, dominios “compatibles”. Para simplificarlo, nosotros supondremos que los dominios deben ser iguales en todos los casos en que, según Date (2001), se aceptarían dominios “compatibles”.

Conviene subrayar que, ...

... tal y como ya hemos mencionado, el modelo relacional permite representar toda la información mediante valores explícitos que contienen las relaciones, y no le hace falta nada más. De este modo, las conexiones entre tuplas de las relaciones se expresan con los valores explícitos de las claves foráneas de las relaciones, y no son necesarios conceptos adicionales (por ejemplo, apuntadores entre tuplas), para establecer estas conexiones. Esta característica da simplicidad y uniformidad al modelo.

Ejemplo de atributo que forma parte de la clave primaria y de una clave foránea

Puede suceder que algún atributo de una relación forme parte tanto de la clave primaria como de una clave foránea de la relación. Esto se da en las relaciones siguientes: EDIFICIOS(nombreedificio, dirección), y DESPACHOS(edificio, número, superficie), donde {edificio} es una clave foránea que referencia EDIFICIOS.

En este ejemplo, el atributo edificio forma parte tanto de la clave primaria como de la clave foránea de la relación DESPACHOS.

2.6. Creación de las relaciones de una base de datos

Hemos visto que una base de datos relacional consta de varias relaciones. Cada relación tiene varios atributos que toman valores de unos ciertos dominios; también tiene una clave primaria y puede tener una o más claves foráneas. Los lenguajes de los SGBD relacionales deben proporcionar la forma de definir todos estos elementos para crear una base de datos.