

Base de Datos

5. El Modelo Relacional



Curso 2021-2022

Tabla de contenido

1. MODELO RELACIONAL.....	3
1.1. ESTRUCTURACIÓN DE LOS DATOS.....	3
1.1.1. Dominio.....	4
1.1.2. Esquema y extensión	5
1.1.3. Claves candidatas, clave primaria y claves alternativas	7
1.1.4. Claves foráneas.....	9
1.1.5. Operaciones con relaciones	11
1.2. REGLAS DE INTEGRIDAD	12
1.2.1. Unicidad de la clave primaria	13
1.2.2. Entidad de la clave primaria	13
1.2.3. Integridad referencial	14
1.2.4. Integridad del dominio	18
1.3. TRADUCCIÓN DEL MODELO ENTIDADRELACIÓN EN EL MODELO RELACIONAL	19
1.3.1. Entidades	20
1.3.2. Interrelaciones	21
1.3.3. Entidades débiles	30
1.3.4. Generalización y especialización	30
1.3.5. Entidades asociativas	31

1. MODELO RELACIONAL

El modelo relacional es un modelo de datos basado en dos disciplinas matemáticas: la lógica de predicados y la teoría de conjuntos.

SGBD

Acrónimo de Sistema Gestor de Bases de Datos. Es un software especializado en la gestión de bases de datos (entendidas éstas como un conjunto estructurado de información).

Quizás debido a este sólido fundamento teórico, que proporciona a este modelo una robustez excepcional, los SGBD relacionales (o SGBDR) son actualmente los que tienen una mayor implantación en el mercado.

El modelo relacional fue propuesto originalmente por Edgar Frank Codd en su trabajo *A Relational Model of Data for Large Shared Data Banks* ('Un modelo relacional de datos para grandes bancos de datos compartidos') en 1970, aunque no se implementó comercialmente hasta finales de la década.

EF Codd

Codd trabajaba para IBM, pero no fue esta multinacional la que creyó antes en las posibilidades del modelo relacional, sino más bien la competencia, y muy especialmente Oracle, empresa que nació, justamente, con el nombre de Relational Software.

1.1. ESTRUCTURACIÓN DE LOS DATOS

El **modelo relacional** permite construir estructuras de datos para representar las distintas informaciones del mundo real que tengan algún interés.

Las estructuras de datos construidas siguiendo el modelo relacional están formadas por conjuntos de relaciones.

Las **relaciones** pueden ser concebidas como representaciones tabulares de los datos.

Tupla

En el ámbito de las BD, podemos definir tupla como una secuencia finita de objetos que comprende las diferentes asociaciones entre cada atributo de la relación y un valor concreto, admisible dentro del respectivo dominio.

Hay que precisar los siguientes extremos:

- Toda relación debe tener un nombre que la identifique unívocamente dentro de la base de datos.
- Cada fila está constituida por un tupla de datos relacionados entre sí, llamado también *registro*, que guarda los datos que nos interesa reflejar de un objeto concreto del mundo real.
- En cambio, cada columna contiene, en cada celda, datos de un mismo tipo, y se la puede llamar *atributo* o *campo*.

Cada celda, o intersección entre fila y columna, puede almacenar un único valor.

Ejemplo de relación

La **tabla 1** refleja la estructuración tabular de la relación ALUMNO, conteniendo los datos personales correspondientes a los individuos matriculados en un centro docente.

Cada fila contiene varios datos relacionados que, en este caso, son los que pertenecen a un mismo alumno.

La relación tiene un nombre (ALUMNO), como cada una de las columnas (DNI, Nombre, Apellidos y Teléfono). Si estos nombres son suficientemente significativos, permiten captar enseguida el sentido que tienen los valores de los datos almacenados en la relación.

ALUMNO			
DNI	Nombre	Apellidos	Teléfono
47126654F	José	Bel Rovira	453641282
51354897S	Anna	Pacheco Cuscó	723352151
56354981L	Javier	Ríos Montalvo	726922235

Tabla 1 Ejemplo de Relación

Toda base de datos relacional está formada por un conjunto de relaciones.

Esta sencilla forma de visualizar la estructura de las bases de datos relacionales resulta muy comprensible para la mayoría de usuarios. Pero hay que profundizar en algunas características adicionales de las relaciones, para poder distinguirlas claramente de los ficheros tradicionales.

1.1.1. Dominio

Por lo que respecta al modelo relacional, un **dominio** consiste en un conjunto finito de valores indivisibles.

Los atributos sólo pueden tomar los valores que estén incluidos dentro del dominio respectivo. De lo contrario, no son valores válidos, y un SGBD relacional no puede permitir su almacenamiento.

Ejemplos de dominios

Examinamos el atributo Telefono de la relación ALUMNO. Si lo definimos de tal forma que sólo pueda almacenar nueve caracteres (porque los teléfonos siempre constan de nueve dígitos) de tipo numérico (ya que las letras no pueden formar parte de un número de teléfono), el dominio de este atributo incluirá todas las combinaciones posibles (en concreto, 10^9 , que es una magnitud grande, pero finita).

Otra cosa es que muchos de estos valores nunca podrán corresponderse con valores existentes en el mundo real (por ejemplo, difícilmente un operador asignará a uno de sus abonados una cadena de nueve ceros como identificador telefónico). Para conseguirlo, habría que restringir mucho más el dominio del atributo a la hora de definirlo.

Centrémonos ahora en el atributo Apellidos. Contendrá los valores de ambos apellidos de los alumnos que los tengan, separados por un espacio en blanco. Por tanto, este campo está definido para que pueda almacenar dos objetos del mundo real: primer apellido y segundo apellido.

Conceptualmente, los usuarios podrán distinguir entre los dos objetos representados, y los programadores de aplicaciones podrán trincar, en caso necesario, el resultado obtenido al realizar una consulta del campo Apellidos. Pero todo SGBD relacional considerará el valor contenido en el atributo Apellidos de forma atómica, sin estructuración interna alguna.

Debemos considerar dos tipologías de dominios:

- **Dominios predefinidos** . Son los tipos de datos que admita cada SGBD, como, por ejemplo (mencionados de forma genérica, ya que existen muchas especificidades en función de los diferentes sistemas gestores), las cadenas de caracteres, los números enteros, los números decimales, los datos de cariz cronológico, etc.
- **Dominios definidos por los usuarios** . Consisten en restricciones adicionales aplicadas sobre el dominio predefinido de algunos atributos, establecidas por los diseñadores y administradores de bases de datos.

Ejemplo de dominio definido por el usuario

En una relación para almacenar los datos de los aspirantes a mozo de escuadra, se podría establecer el campo IMC, para registrar los respectivos índices de masa corporal.

Pues bien, se podría restringir el dominio de este campo de tal forma que no admitiera aspirantes con valores inferiores a diecinueve ni superiores a treinta, puesto que la normativa no lo permite.

1.1.2. Esquema y extensión

Toda relación consta de un esquema y de su extensión.

El **esquema de una relación** consiste en un nombre que la identifica unívocamente dentro de la base de datos, y en el conjunto de atributos que aquella contiene.

Es muy recomendable, para evitar confusiones en la ulterior implementación, seguir uniformemente una notación concreta a la hora de expresar los esquemas de las relaciones que forman una misma base de datos.

A continuación, se detallan las características de uno de los sistemas de notación más frecuentes:

- Es necesario escribir el nombre de las relaciones con mayúsculas y preferiblemente en singular.
- Debe escribirse el nombre de los atributos empezando con mayúscula y continuando con minúsculas, siempre que no se trate de siglas, ya que entonces es más conveniente dejar todas las letras con mayúsculas (como DNI). Para realizar los nombres compuestos más leedores, se puede encabezar cada palabra de las que forman el nombre del campo con una letra mayúscula (por ejemplo: DataNacimiento, Telefono Particular, etc.).

Ejemplo de esquema de una relación

El esquema de la relación que se muestra en la **tabla 2** , conforme al sistema de notación propuesto, quedaría como sigue:

ALUMNO(DNI, Nombre, Apellidos, Teléfono)

Hay que precisar que el orden en el que nos muestren los atributos es indiferente, por definición del modelo relacional.

ALUMNO			
DNI	Nombre	Apellidos	Teléfono
47126654F	José	Bel Rovira	453641282
51354897S	Anna	Pacheco Cuscó	723352151
56354981L	Javier	Ríos Montalvo	726922235

Tabla 2 Ejemplo de relación

Los atributos de una relación son únicos dentro de ésta. Su nombre no puede estar repetido dentro de una misma relación. Ahora bien, distintas relaciones sí pueden contener atributos con el mismo nombre.

Por otro lado, cabe decir que los dominios de diferentes atributos de una misma relación pueden ser idénticos, aunque los respectivos campos almacenen los valores de diferentes propiedades del objeto (por ejemplo, sería perfectamente lógico que los atributos TelefonFijo, TelefonoMovil y TelefonoTrabajo, a pesar de pertenecer a una misma relación, tuvieran el mismo dominio).

*La **extensión de una relación** consiste en los valores de los datos almacenados en todas las tuplas que ésta contiene.*

Ejemplo de extensión

Si tomamos como base, una vez más, la relación con esquema ALUMNO(DNI, Nombre, Apellidos, Teléfono) de la tabla 2, su extensión sería una lista en la que figurarían todos los alumnos de la base de datos:

Alumno 1: 47126654F, José, Bel Rovira, 453641282 Alumno 2: 51354897S, Anna, Pacheco Cuscó, 723352151 Alumno 3: 56354981L, Xavier, Rius Montalvo2

En ocasiones, los atributos de las relaciones pueden no contener valor alguno o, dicho de otro modo, pueden contener valores nulos.

Ejemplo de valor nulo

Imaginemos que se matricula un cuarto alumno que no tiene teléfono. Sus datos en la conocida relación con esquema ALUMNO (DNI, Nombre, Apellidos, Telefono) reflejarán esta circunstancia con la ausencia de valor en el atributo Telefono del tupla que le corresponda.

Al utilizar representaciones tabulares para visualizar los valores de las extensiones de las relaciones (en el plano teórico, no en implementaciones reales con SGBD), para indicar que una celda tiene valor nulo se puede incluir la palabra NUL (como en la *tabla 3*), o bien se puede dejar en blanco, simplemente.

ALUMNO			
DNI	Nombre	Apellidos	Teléfono
47126654F	José	Bel Rovira	453641282

51354897S	Anna	Pacheco Cuscó	723352151
56354981L	Javier	Ríos Montalvo	726922235
24583215W	Mariona	Castellví Mur	NULO

Tabla 3 Ejemplo de relación con valores nulos

El **grado** de una relación depende del número de atributos que incluye su esquema.

Ejemplo de grado de una relación

La relación con esquema ALUMNO (DNI, Nombre, Apellidos, Teléfono) de la **tabla 3** es de grado 4, por tener cuatro atributos.

*La **cardinalidad** de una relación viene dada por el número de tuplas que forman la extensión.*

Ejemplo de cardinalidad

Si nos fijamos en la **tabla 3**, la cardinalidad de la relación ALUMNO es 4, porque su extensión contiene cuatro tuplas correspondientes a los cuatro alumnos que, de momento, están matriculados.

1.1.3. Claves candidatas, clave primaria y claves alternativas

Con el fin de resultar útil, el almacenamiento de la información debe permitir la identificación de los datos. En el ámbito de las bases de datos relacionales, los tuplas de las relaciones se identifican mediante las llamadas *superllaves*.

Una **superclave** es un subconjunto de los atributos que forman el esquema de una relación tal que no es posible que haya más de un tupla en su respectiva extensión, con la misma combinación de valores en los atributos que forman parte de dicho subconjunto.

Pero una superclave puede contener atributos innecesarios, que no contribuyen a la identificación inequívoca de los distintos tuplas. Lo que habitualmente interesa es trabajar con **superclaves mínimas**, tales que ningún subconjunto propio sea capaz por sí solo de identificar los tuplas de la relación.

Por definición, ninguna superclave mínima podrá admitir valores nulos en ninguno de sus atributos, porque si lo hiciera, podría garantizar la identificación inequívoca de los tuplas que contuvieran algún valor nulo en algunos de los atributos de la superclave mínima en cuestión.

Por otra parte, cabe decir que en una misma relación puede ocurrir que haya más de una superclave mínima que permita distinguir a los tuplas unívocamente entre ellos.

Se llaman **claves candidatas** a todas las superclaves mínimas de una relación formadas por los atributos o conjuntos de atributos que permiten identificar a los tuplas que contiene su extensión.

Elección de la clave primaria

Con mucha frecuencia, es el administrador de la BD quien elige la clave primaria de la relación, entre las claves candidatas disponibles, realizando, en el fondo, tareas de diseñador lógico.

Sin embargo, a la hora de implementar una BD, entre todas las claves candidatas de cada relación sólo se debe elegir una.

Cuando hablamos de **clave primaria** nos referimos a la clave que, finalmente, el diseñador lógico de la base de datos elige para distinguir unívocamente a cada tupla de una relación del resto.

Entonces, las claves candidatas no elegidas como clave primaria están presentes en la relación.

Cuando una relación ya tiene establecida una clave primaria, el resto de claves presentes en aquella, y que también podrían servir para identificar los diferentes tuplas de la extensión respectiva, se conocen como **claves alternativas**.

Una forma de diferenciar los atributos que forman la clave primaria de las relaciones, de los otros atributos del respectivo esquema, es ponerlos subrayados. Por este motivo, normalmente se colocan juntos y antes que el resto de atributos, dentro del esquema. Pero sólo se trata de una cuestión de elegancia, puesto que el modelo relacional no se basa ni en el orden de los atributos del esquema, ni tampoco en el orden de los tuplas de la extensión de la relación.

Ejemplos de claves candidatas, primaria y alternativas

Observando la **tabla 4**, podemos imaginar que la relación ALUMNO tiene varios atributos más, de modo que su esquema queda como sigue:

ALUMNO(DNI, NumSS, NumMatricula, Nombre, Apellidos, Teléfono)

Veremos fácilmente cómo los atributos DNI, NumSS (número de la Seguridad Social) y NumMatricula, al ser personales e irrepetibles, podrían servirnos para identificar unívocamente a los alumnos. Por tanto, serían claves candidatas.

Entonces, el diseñador de BD deberá decidirse por una clave candidata como clave primaria. Si, por ejemplo, elige DNI como clave primaria, las antiguas claves candidatas restantes se pasarán a considerar claves alternativas.

En este caso, pues, el esquema resultante deberá reflejar cuál es la clave primaria de la relación, subrayando el atributo DNI:

ALUMNO(DNI , NumSS, NumMatricula, Nombre, Apellidos, Teléfono)

ALUMNO			
DNI	Nombre	Apellidos	Teléfono
47126654F	José	Bel Rovira	453641282
51354897S	Anna	Pacheco Cuscó	723352151
56354981L	Javier	Ríos Montalvo	726922235
24583215W	Mariona	Castellví Mur	NULO

Tabla 4 Ejemplo de relación con valores nulos

Si no se dispone de un atributo que sea capaz de identificar los tuplas de la relación por sí solo, es necesario buscar un subconjunto de atributos, tales que la combinación de los valores que adopten no pueda repetirse. Si esta posibilidad no existe, es necesario añadir a la relación un atributo adicional que haga de identificador.

Por definición, el modelo relacional no admite tuplas repetidos, es decir, no permite la existencia de tuplas en una misma relación que tengan los mismos valores en cada uno de los atributos.

Ahora bien, las implementaciones concretas de los distintos SGBD sí permiten esta posibilidad, siempre que no se establezca ninguna clave primaria en la relación con tuplas repetidos.

Esta permisividad a veces permite solucionar ciertas eventualidades, pero no debería ser la forma habitual de trabajar con BD relacionales.

1.1.4. Claves foráneas

Cualquier BD relacional, por pequeña que sea, contiene normalmente más de una relación. Para reflejar correctamente los vínculos existentes entre algunos objetos del mundo real, las tuplas de las diferentes relaciones de una base de datos deben interrelacionarse.

En ocasiones, incluso puede ser necesario relacionar los tuplas de una relación con otros tuplas de la misma relación.

El mecanismo que ofrecen las BD relacionales para interrelacionar las relaciones que contienen se fundamenta en las llamadas *llaves foráneas*.

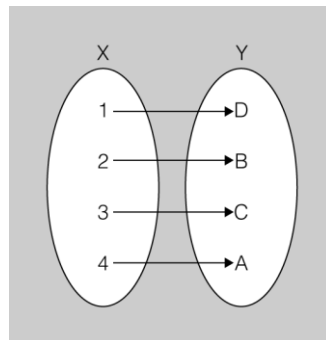
Una **clave foránea** está constituida por un atributo, o por un conjunto de atributos, del esquema de una relación, que sirve para relacionar sus tuplas con los tuplas de otra relación de la base de datos (o con los tuplas de sí misma, en algunos casos).

Con el fin de conseguir conectar los tuplas de una relación con los de otra (o con las suyas propias), la clave foránea utilizada debe referenciar la clave primaria de la relación con la que se quiere relacionar.

Las diferentes combinaciones de valores de los atributos de toda clave foránea deben existir en la clave primaria a la que hacen referencia, o bien deben ser valores nulos. Por otra parte, las referencias serían erróneas y, por tanto, los datos serían incorrectos.

Hay que prestar atención a las siguientes características de las claves foráneas:

- Toda clave foránea debe tener el mismo número de atributos que la clave primaria a la que hace referencia.
- Entre los atributos del esquema de una clave foránea y los de la clave primaria respectiva debe poder establecerse una correspondencia (concretamente, una biyección).
- Los dominios de los atributos de toda clave foránea deben coincidir con los dominios de los atributos de la clave primaria respectiva (o, como mínimo, deben ser compatibles dentro de un cierto rango).



Una relación puede contener más de una clave foránea, o bien no contener ninguna. Y, en sentido inverso, la clave primaria de una relación puede estar referenciada por una o más claves foráneas, o bien puede no estar referenciada por ninguna.

Por último, cabe decir que puede darse el caso de que un mismo atributo forme parte tanto de la clave primaria de la relación como de alguna de sus claves foráneas.

Ejemplos de claves foráneas

La relación ALUMNO, tal y como se muestra en la tabla 5 , incorpora dos claves foráneas.

Una de ellas, CodiAula, hace referencia a la clave primaria de la relación AULA (formada por el atributo Codi), expuesta en la tabla 6 , para indicar qué aula corresponde a cada alumno.

En cambio, DNIDelegado hace referencia a la clave primaria de la misma relación (formada por el atributo DNI), y sirve para indicar cuál es el delegado que representa a cada alumno.

Fijémonos en que la alumna Mariona Castellví todavía no tiene asignado ni delegado ni aula y, por este motivo, el tuplo que la representa contiene, de momento, valores nulos en los atributos de las dos claves foráneas.

ALUMNO					
DNI	Nombre	Apellidos	Teléfono	DNIDelegado	CódigoAula
51354897S	Anna	Pacheco Cuscó	723352151	51354897S	201
56354981L	Javier	Ríos Montalvo	726922235	51354897S	201
24583215W	Mariona	Castellví Mur	NULO	NULO	NULO

Tabla 5 Ejemplo de relación con valores nulos

AULA	
Código	Capacidad
101	40
201	30

Tabla 6 Ejemplo de relación con clave primaria referenciada

La notación más habitual para designar las claves foráneas de las relaciones consiste en añadir esta circunstancia a continuación de su esquema, incluyendo entre dos claves el conjunto de atributos que forman la clave foránea de que se trate, precedido del adverbio **ON** , y seguido de la forma verbal **REFERENCIA** y de la relación a la que

hace referencia. Si hay más de una llave foránea, se separan por comas y la última debe ir precedida de la conjunción *y*.

EJEMPLO DE NOTACIÓN POR DESIGNAR CLAVES FORÁNEAS

Las dos relaciones que se muestran en las tablas 5 y 6 se expresarán de la siguiente forma:

ALUMNO(DNI , Nombre, Apellidos, Teléfono, DNIDelegado, CódigoAula) DONDE {DNIDelegado} REFERENCIA ALUMNO y {CodigoAula} REFERENCIA AULA

AULA(Código , Capacidad)

1.1.5. Operaciones con relaciones

El modelo relacional permite realizar una serie de operaciones con los datos almacenados en las BD, que tienen diferentes finalidades:

- **Actualización.** Estas operaciones realizan cambios en los tuplas que quedan reflejados en las relaciones contenidas en las BD. Pueden ser de tres tipos:
 - **Inserción.** Consiste en añadir uno o más tuplas nuevos a una relación determinada.
 - **Eliminación.** Consiste en eliminar uno o más tuplas nuevos de una relación determinada.
 - **Modificación.** Consiste en cambiar el valor de uno o más atributos de uno o más tuplas de una determinada relación.
- **Consulta.** Estas operaciones sólo hacen posible la obtención parametrizada de datos, sin que se vean alteradas las almacenadas en la BD.

La realización de estas operaciones comporta el previo conocimiento de la estructura formada por las relaciones que sea necesario utilizar, es decir, los esquemas de las relaciones y las interrelaciones entre ellas, mediante las claves foráneas.

Ejemplos de operaciones

Si tomamos en consideración la relación ALUMNO que muestra la tabla 7 , un ejemplo de inserción consistirá en añadir un nuevo alumno, como el siguiente:

<65618724G, Lidia, Bofarull Mora, 564628231, 47126654F, 102>

Un ejemplo de borrado sería eliminar el tupla que contiene los datos de un alumno dado de alta, como el siguiente:

<56354981L, Xavier, Rius Montalvo, 726922235, 51354897S, 201>

Un ejemplo de modificación sería, por ejemplo, cambiar el número de teléfono de Josep Bel Rovira que consta en la BD (453641282) por otro (546022547), para reflejar correctamente la realidad, de forma actualizada, o asignarla uno nuevo a alguien que antes no tenía, como Mariona Castellví Mur, introduciendo 875261473 en lugar del anterior valor nulo.

Y, como ejemplo de consulta, podría interesarnos obtener una lista, ordenada alfabéticamente por los apellidos, de todos los alumnos que son delegados de

aula y que, por tanto, tienen valores coincidentes en el atributo DNI y en el atributo DNIDelegado . En este caso, el resultado sería el siguiente:

<47126654F, José, Bel Rovira, 546022547, 47126654F, 102>

<51354897S, Anna, Pacheco Cuscó, 723352151, 51354897S, 201>

ALUMNO					
DNI	Nombre	Apellidos	Teléfono	DNIDelegado	CódigoAula
47126654F	José	Bel Rovira	453641282	47126654F	102
51354897S	Anna	Pacheco Cuscó	723352151	51354897S	201
56354981L	Javier	Ríos Montalvo	726922235	51354897S	201
24583215W	Mariona	Castellví Mur	NULO	NULO	NULO

Tabla 7 Ejemplo de relación con claves foráneas

1.2. REGLAS DE INTEGRIDAD

Los valores que almacenan las BD deben reflejar en todo momento, de forma correcta, la porción de la realidad que queremos modelizar.

Llamamos **integridad** la propiedad de los datos que consiste en representar correctamente las situaciones del mundo real que modelizan.

Para que los datos sean íntegros, es necesario garantizar que sean correctos, y también que estén enteros.

En atención al objetivo mencionado, pues, los datos deben cumplir ciertas condiciones, que podemos agrupar en dos tipologías distintas:

- **Restricciones de integridad del usuario** . Son condiciones específicas de cada BD. Los SGBD permitirán a los administradores establecer ciertas restricciones aplicables a casos concretos, garantizando que se respeten durante la explotación habitual del sistema.
- **Reglas de integridad del modelo** . Son condiciones de carácter general que deben cumplir todas las BD que sigan el modelo relacional. No es necesario definirlas al implementar cada BD, porque se consideran preestablecidas.

Ejemplo de restricción de integridad del usuario

Al dar de alta un nuevo alumno de la relación ALUMNO, que se muestra en la tabla 8, podríamos exigir al sistema que lo validase, mediante el algoritmo correspondiente, si la letra introducida del NIF se corresponde con las cifras previamente introducidas, y que denegara la inserción de lo contrario, a fin de no almacenar una situación en principio no admisible en el mundo real.

Ejemplo de regla de integridad del modelo

Dado que la relación ALUMNO, que se muestra en la tabla 8 , tiene definido el atributo DNI como clave primaria, el sistema validará automáticamente que no se introduzca más

de un alumno con el mismo carné de identidad, ya que entonces la clave primaria no cumpliría su objetivo de garantizar la identificación inequívoca de cada tupla , diferenciándolo del resto.

ALUMNO					
DNI	Nombre	Apellidos	Teléfono	DNIDelegado	CódigoAula
47126654F	José	Bel Rovira	453641282	47126654F	102
51354897S	Anna	Pacheco Cuscó	723352151	51354897S	201
56354981L	Javier	Ríos Montalvo	726922235	51354897S	201
24583215W	Mariona	Castellví Mur	NULO	NULO	NULO

Tabla 8 Ejemplo de relación con claves foráneas

1.2.1. Unicidad de la clave primaria

El valor de una clave primaria, globalmente considerada, no puede estar repetido en más de un tupla de la misma relación, ya que entonces la clave primaria no estaría en condiciones de asegurar la identificación inequívoca de los distintos tuplas.

En ningún momento, puede haber dos o más tuplas con la misma combinación de valores en el conjunto de los atributos que forman la clave primaria de una relación.

Los SGBD relacionales deben garantizar la regla de unicidad de la clave primaria en todas las inserciones de nuevos tuplas, así como en todas las modificaciones que afecten al valor de alguno de los atributos que formen parte de la clave primaria.

Ejemplo de unicidad de la clave primaria

En la relación AULA, que se muestra en la tabla 9 , no debería poder insertarse un nuevo tupla con los valores **<102, 40>**, porque la clave primaria ya almacena el valor 102, correspondiente a otro tupla .

Si queremos dar de alta otra aula en el primer piso del edificio con capacidad para cuarenta alumnos, deberemos utilizar como clave primaria otro valor no presente en el atributo Código, resultando, por ejemplo, **<103, 40>**.

Tampoco debería ser posible modificar la clave del tuplo **<101, 40>** y asignarle el valor **102**, porque este valor ya lo tiene asignado la clave primaria de otro tupla .

AULA	
Código	Capacidad
101	40
102	36
201	30

Tabla 9 Ejemplo de relación con clave primaria referenciada

1.2.2. Entidad de la clave primaria

Las claves primarias sirven para diferenciar cada tuplo de una relación del resto de tuplas de la misma relación. Para garantizar la consecución de esta finalidad, es necesario que los atributos que forman parte de una clave primaria no puedan tener

valor nulo ya que, si se admitiera esta posibilidad, los tuplas con valores nulos en la clave primaria no podrían distinguirse de algunos otros.

Ningún atributo que forme parte de una clave primaria puede contener nunca valores nulos en ningún tupla .

Los SGBD relacionales deben garantizar la regla de entidad de la clave primaria en todas las inserciones de nuevas tuplas, así como en todas las modificaciones que afecten al valor de alguno de los atributos que formen parte de la clave primaria.

Ejemplo de entidad de la clave primaria

En la relación AULA, que se muestra en la tabla 10 , no debería poder insertarse un nuevo tupla con los valores **<NUL, 26>**, porque la clave primaria, por definición, no puede contener valores nulos.

Si queremos dar de alta otra aula con capacidad para veintiséis alumnos, deberemos utilizar como clave primaria otro valor no nulo, resultando, por ejemplo, **<202, 26>**.

Tampoco debería ser posible, por la misma razón que hemos expuesto anteriormente, modificar la clave del tupla **<101, 40>** y asignarle el valor nulo.

AULA	
Código	Capacidad
101	40
102	36
201	30

Tabla 10 Ejemplo de relación con clave primaria referenciada

1.2.3. Integridad referencial

El modelo relacional no admite, por definición, que la combinación de valores de los atributos que forman una clave foránea no esté presente en la clave primaria correspondiente, puesto que esto implicaría una conexión incorrecta.

La **integridad referencial** implica que, para cualquier tupla, la combinación de valores que adopta el conjunto de los atributos que forman la clave foránea de la relación o bien debe estar presente en la clave primaria a la que hace referencia, o bien debe estar constituida exclusivamente por valores nulos (si los atributos implicados admiten esta posibilidad, y así se ha estipulado al definir sus propiedades).

Los SGBD relacionales deberán realizar las comprobaciones pertinentes, de forma automática, para garantizar la integridad referencial, cuando se produzcan dos tipos de operaciones con relaciones que tengan claves foráneas:

- Inserciones de nuevas tuplas.
- Modificaciones que afecten a atributos que formen parte de cualquier clave foránea.

Por otra parte, los SGBD relacionales también tendrán que validar la corrección de otros dos tipos de operaciones con relaciones que tengan la clave primaria referenciada desde alguna clave foránea:

- Borramiento de tuplas.

- Modificaciones que afecten a atributos que formen parte de la clave primaria.

Para garantizar la integridad referencial en estos dos últimos tipos de operación, se puede seguir alguna de las tres políticas siguientes: restricción, actualización en cascada y anulación.

Ejemplo de violación de la integridad referencial

Continuamos especulando con las relaciones ALUMNO y AULA (reflejadas en la tabla 11 y tabla 12 respectivamente).

El tupla que contiene los datos de Mariona Castellví Mur tiene un valor nulo en el atributo que forma la clave foránea que hace referencia a la relación AULA.

Si lo quisiéramos actualizar con el valor **316**, por ejemplo, el sistema no debería dejarlo hacerlo, porque este valor no está presente en la clave primaria de ningún tupla de la relación AULA y, por tanto, esta operación contravendría la regla de integridad referencial.

ALUMNO					
DNI	Nombre	Apellidos	Teléfono	DNIDelegado	CódigoAula
47126654F	José	Bel Rovira	453641282	47126654F	102
51354897S	Anna	Pacheco Cuscó	723352151	51354897S	201
56354981L	Javier	Ríos Montalvo	726922235	51354897S	201
24583215W	Mariona	Castellví Mur	NULO	NULO	NULO

Tabla 11 Relación con llaves foráneas

AULA	
Código	Capacidad
101	40
102	36
201	30

Tabla 12 Relación con clave primaria referenciada

Política de restricción

La política de restricción consiste en prohibir la operación de actualización de que se trate:

- En caso de borrado, no permitirá eliminar un tupla si su clave primaria está referenciada desde alguna clave foránea.
- En caso de modificación, no permitirá alterar el valor de ninguno de los atributos que forman la clave primaria de un tupla, si ésta está referenciada desde alguna clave foránea.

Ejemplos de restricciones

Consideramos una vez más las relaciones ALUMNO y AULA que se muestran en la tabla 13 y tabla 14 respectivamente.

Aplicando la restricción tanto en caso de borrado como de modificación, estas operaciones no serán posibles con el aula **102** de la relación AULA porque hay alumnos matriculados que deben asistir a clase dentro de este espacio y, por tanto, la referencian desde la clave foránea de los tuplas que los representan. Sí sería posible, en cambio, borrar el aula 101, puesto que no está referenciada desde la relación ALUMNO.

ALUMNO					
DNI	Nombre	Apellidos	Teléfono	DNIDelegado	CódigoAula
47126654F	José	Bel Rovira	453641282	47126654F	102
51354897S	Anna	Pacheco Cuscó	723352151	51354897S	201
56354981L	Javier	Ríos Montalvo	726922235	51354897S	201
24583215W	Mariona	Castellví Mur	NULO	NULO	NULO

Tabla 13 Relación con llaves foráneas

AULA	
Código	Capacidad
101	40
102	36
201	30

Tabla 14 Relación con clave primaria referenciada

Actualización en cascada

La política de actualización en cascada consiste en permitir la operación de actualización de que se trate sobre un tupla determinado, pero disponiendo al mismo tiempo una serie de operaciones compensatorias que propaguen en cascada las actualizaciones necesarias para que se mantenga la integridad referencial de los tuplas que referencian, desde los atributos que forman la clave foránea, el tupla objeto de actualización:

- En caso de borrado, se eliminarán todos los tuplas que hagan referencia al tupla borrado.
- En caso de modificación, los valores de los atributos que formen parte de la clave foránea de los tuplas que hagan referencia al tupla modificado se alterarán para continuar coincidiendo con los nuevos valores de la clave primaria del tupla al que se refieren.

Ejemplos de actualización en cascada

Volvemos a tomar como punto de partida de los ejemplos las relaciones ALUMNO y AULA que se muestran en la tabla 15 y tabla 16 respectivamente.

Si aplicamos la actualización en cascada borrando el tuplo **<201, 30>** de la relación AULA, también se borrarán los dos tuplas de la relación ALUMNO que hacen referencia a ella desde la clave foránea respectiva (CodiAula).

En cambio, si aplicamos la actualización en cascada modificando el tupla **<201, 30>** de la relación AULA, cambiando el valor de su clave primaria por otro, como **203**, los dos

tuplas de la relación ALUMNO que hacen referencia actualizarán en cascada el valor del atributo CodiAula de **201** a **203**, a fin de mantener la conexión correcta entre los tuplas de ambas relaciones.

ALUMNO					
DNI	Nombre	Apellidos	Teléfono	DNIDelegado	CódigoAula
47126654F	José	Bel Rovira	453641282	47126654F	102
51354897S	Anna	Pacheco Cuscó	723352151	51354897S	201
56354981L	Javier	Ríos Montalvo	726922235	51354897S	201
24583215W	Mariona	Castellví Mur	NULO	NULO	NULO

Tabla 15 Relación con llaves foráneas

AULA	
Código	Capacidad
101	40
102	36
201	30

Tabla 16 Relación con clave primaria referenciada

Política de anulación

La política de anulación consiste en permitir la operación de actualización de que se trate en un tupla determinado, pero disponiendo al mismo tiempo una serie de operaciones compensatorias que pongan valores nulos en todos los atributos que formen parte de las claves foráneas de los tuplas que hagan referencia al tupla objeto de actualización:

- En caso de borrado, los atributos de la clave foránea de los tuplas que hagan referencia al tupla borrado pasarán a tener valor nulo, sin indicar ningún tipo de conexión.
- En caso de modificación, los atributos de la clave foránea de los tuplas que hagan referencia al tupla modificado pasarán a tener valor nulo, no indicando ningún tipo de conexión.

La política de anulación sólo puede aplicarse si los atributos de las claves foráneas implicadas admiten los valores nulos.

Ejemplo de anulación

Tomamos una vez más como punto de partida de los ejemplos las relaciones ALUMNO y AULA que se muestran en la tabla 17 y tabla 18 respectivamente.

Aplicando la política de anulación, tanto si borramos el tupla **<102, 36>** de la relación AULA, como si sólo cambiamos el valor del atributo de su clave primaria (Código) por otro (como, por ejemplo, **105**), el tupla de la relación ALUMNO que hace referencia actualizará el valor del atributo CódigoAula de **102** a valor nulo, a fin de evitar una conexión incorrecta entre los tuplas de ambas relaciones, resultando entonces el tupla siguiente:

<47126654F, Josep, Bel Rovira, 453641282, 47126654F, NULL>

ALUMNO					
DNI	Nombre	Apellidos	Teléfono	DNIDelegado	CódigoAula
47126654F	José	Bel Rovira	453641282	47126654F	NULO
51354897S	Anna	Pacheco Cuscó	723352151	51354897S	201
56354981L	Javier	Ríos Montalvo	726922235	51354897S	201
24583215W	Mariona	Castellví Mur	NULO	NULO	NULO

Tabla 17 Relación con llaves foráneas

AULA	
Código	Capacidad
101	40
201	30

Tabla 18 Relación con clave primaria referenciada

Selección de la política a seguir

Será el diseñador de cada BD quien escoja la política más adecuada a seguir en cada caso concreto. Como orientación, conviene saber que las opciones más frecuentes, siempre que no sea necesario realizar consideraciones adicionales, son las siguientes:

- En caso de borrado, normalmente se opta por la restricción.
- En caso de modificación, lo habitual es optar por la actualización en cascada.

La política de anulación es mucho menos frecuente, y se pone en práctica cuando quieren conservarse ciertos datos, aunque hayan perdido la conexión que tenían antes, a veces con la esperanza de que la puedan recuperar más adelante.

1.2.4. Integridad del dominio

La regla de integridad del dominio implica que todos los valores no nulos contenidos en los atributos de las relaciones de cualquier BD deben pertenecer a los respectivos dominios declarados para los atributos en cuestión.

Esta condición es aplicable tanto en lo que se refiere a los dominios predefinidos, como también en lo que se refiere a los dominios definidos por el usuario.

La regla de integridad del dominio comporta también que los operadores que es posible aplicar sobre los valores dependen de los dominios de los respectivos atributos que los almacenan.

Ejemplos de integridad de dominio

Fijémonos, por última vez, en la relación AULA que se muestra en la tabla 19 .

AULA	
Código	Capacidad

101	40
102	36
201	30

Tabla 19 Relación con clave primaria referenciada

Si en la relación con esquema AULA(Código, Capacidad) definimos el dominio del atributo Código como el de los números enteros de 0 hasta 999, entonces no podremos insertar, por ejemplo, un valor en el atributo que forma la clave primaria que no pertenezca a su dominio, tales como INF, o LAB.

Tampoco podremos aplicar determinados operadores para comparar valores de la clave primaria con valores que no pertenezcan a su dominio. Así, no podremos consultar las características de un aula con Código='INF', puesto que 'INF' es una cadena de caracteres.

1.3. TRADUCCIÓN DEL MODELO ENTIDADRELACIÓN EN EL MODELO RELACIONAL

Modelo Entidad Relación o modelo ER

El modelo ER es un modelo de datos cuyo resultado es un diagrama ER o diagrama Chen donde gráficamente se pueden identificar los principales elementos de datos, sus características más importantes y las interrelaciones entre los mismos.

Una vez conocidas las características de un modelo de bases de datos relacional, será necesario partir del modelo conceptual general (del modelo Entidad-Relación) y realizar un estudio del diseño lógico de bases de datos en este ámbito, el relacional.

En todos los ejemplos, presupondremos que previamente ha tenido lugar una fase de diseño conceptual de la que ha resultado un modelo EntidadRelación (o modelo ER) recogido en los diagramas Chen de que se trate en cada caso.

Fases del diseño de BD

Las fases del diseño de BD son:

1. Diseño conceptual
2. Diseño lógico
3. Diseño físico

Antes de implementar propiamente la BD dentro del entorno ofrecido por el SGBD utilizado, es necesario transformar estos diagramas en estructuras de datos relacionales.

El modelo ER se basa en las entidades y las interrelaciones existentes entre ellas. Podemos avanzar algunos aspectos generales sobre cómo traducir estos elementos al modelo relacional:

- Las entidades siempre dan lugar a relaciones, sean del tipo que sean (a excepción de las entidades auxiliares tipo DATA).
- Las interrelaciones binarias de conectividad 1-1 o 1-N originan claves foráneas en relaciones ya existentes.

- Las interrelaciones binarias de conectividad M-N y todas las n-arias de orden superior a 2 siempre se transforman en nuevas relaciones.

Es conveniente seguir un cierto orden a la hora de diseñar lógicamente una base de datos. Una buena práctica puede consistir en proceder de la siguiente forma:

1. En primer lugar, es necesario transformar las entidades del diagrama con el que trabajamos en relaciones.
2. Luego debe seguir transformándose en relaciones las entidades que presentan algún tipo de especificidad (es decir, las débiles, las asociativas, o las derivadas de un proceso de generalización o especialización).
3. A continuación, deben añadirse a las anteriores relaciones los atributos necesarios para formar las claves foráneas derivadas de las interrelaciones binarias con conectividad 1-1 y 1-N presentes en el diagrama ER.
4. Y, por último, ya puede empezar la transformación de las interrelaciones binarias con conectividad M-N y de las interrelaciones n-arias.

Ningún SGBD puede resolver una referencia a una tabla que todavía no ha sido creada.

De esta forma evitaremos que existan claves foráneas que hagan referencia a relaciones que todavía no se han descrito. Esto hace más leedor el modelo relacional obtenido, ciertamente, pero también ahorra el trabajo de tener que ordenar las relaciones a la hora de escribir (típicamente en lenguaje SQL) y las instrucciones pertinentes para que el SGBD utilizado cree las tablas de la base de datos.

Las técnicas necesarias para realizar correctamente el diseño lógico de bases de datos, según el tipo de conceptualización de que se trate en cada caso.

1.3.1. Entidades

Cada entidad del modelo ER se transforma en una relación del modelo relacional:

- Los atributos de la entidad originaria serán los atributos de la relación resultante.
- La clave primaria de la entidad originaria será la clave primaria de la relación resultante.
- Cuando una entidad interviene en alguna interrelación binaria 1-1 o 1-N, puede ser necesario añadir ulteriormente nuevos atributos, a fin de que actúen como claves foráneas de la relación.

Ejemplo de transformación de entidad

El diagrama ER de la figura 1 se traduce en el modelo relacional de la siguiente forma:

ALUMNO(*DNI* , Nombre, Apellidos)

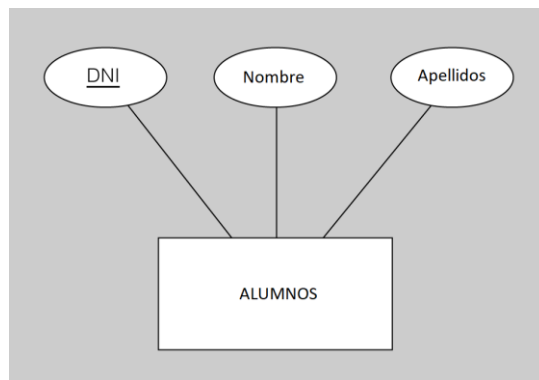


Figura 1

1.3.2. Interrelaciones

Una vez transformadas todas las entidades en relaciones, es necesario traducir las interrelaciones en las que aquéllas participan.

1) **Binarias**. Para traducir las interrelaciones binarias es necesario tener en cuenta su conectividad, así como también las dependencias de existencia.

a) **Conectividad 1-1 y dependencias de existencia**. Es necesario añadir a cualquiera de las dos relaciones una clave foránea que haga referencia a la otra relación.

Pero si una de las dos entidades es opcional en la relación, entonces es ella quien debe acoger la clave foránea, a fin de evitar, de lo contrario, el almacenamiento de valores nulos en la misma, y ahorrarse así espacio de almacenamiento.

Dependencias de existencia

En ocasiones, una entidad instancia sólo tiene sentido si existe al menos otra entidad instancia asociada a ella mediante una interrelación binaria determinada. En estos casos, se dice de la última entidad que es una entidad obligatoria en la interrelación. Por otra parte, se dice que se trata de una entidad opcional en la interrelación.

Los atributos de la interrelación (si los hubiere) acompañan a la clave foránea.

Ejemplo de transformación de interrelación binaria con conectividad 1-1

El diagrama ER de la figura 2 representa una interrelación binaria con conectividad 1-1. Por tanto, en principio habría dos posibilidades de transformación, según se coloca la llave foránea en la entidad PROFESOR o en la entidad DEPARTAMENTO:

DEPARTAMENTO(*Código* , Descripción)

PROFESOR(*DNI*, Nombre, Apellidos, CódigoDepartamento) DONDE {Código Departamento} REFERENCIA DEPARTAMENTO y CódigoDepartamento ADMITE VALORES NULOS

O bien:

PROFESOR(*DNI* , Nombre, Apellidos)

DEPARTAMENTO(*Código*, Descripción, DNIProfesor) DONDE {DNIProfesor} REFERENCIA PROFESOR

Ahora bien, la entidad DEPARTAMENTO es opcional en la interrelación Coordina. Esto significa que puede haber profesores que no coordinen ningún departamento. Por tanto, la opción más correcta consiste en añadir la clave foránea a la relación DEPARTAMENTO, ya que si se añadiera a la relación PROFESOR debería tomar el valor nulo en muchos casos, y ocuparía un espacio de almacenamiento innecesario.

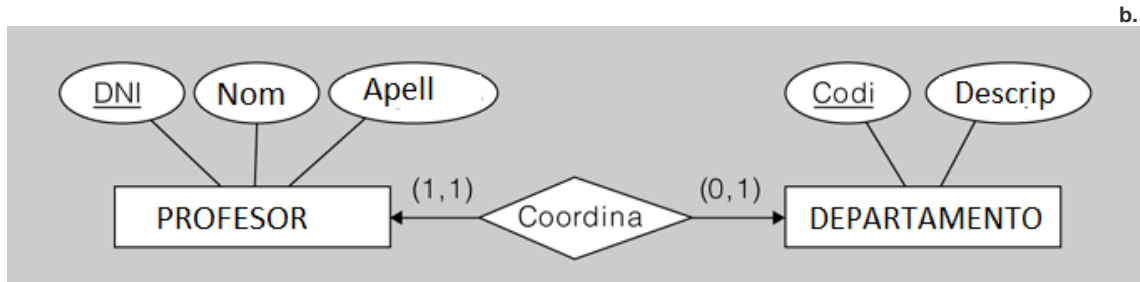


Figura 2 Interrelación binaria con conectividad 1:1

- b) **Conectividad 1-N.** En estos casos se debe añadir una clave foránea a la relación que resulta de traducir la entidad ubicada en el lado N de la interrelación, que haga referencia a la otra relación.

Si se colocara la clave foránea en la otra relación, el atributo que la forma debería ser multivalente para poder representar todas las conexiones posibles, y esto no está permitido dentro del modelo relacional.

Los atributos de la interrelación (si los hubiere) acompañan a la clave foránea.

Ejemplo de transformación de interrelación binaria con conectividad 1-N

El diagrama ER de la figura 3 representa una interrelación binaria con conectividad 1-N. Por tanto, la clave foránea deberá añadirse necesariamente a la entidad derivada de la entidad del lado N, resultando el siguiente modelo:

DEPARTAMENTO(*Código* , Descripción)

PROFESOR(*DNI* , Nombre, Apellidos, CódigoDepartamento) DONDE {Código Departamento} REFERENCIA DEPARTAMENTO y CódigoDepartamento ADMITE VALORES NULOS

La entidad del lado 1 (DEPARTAMENTO) es opcional en la interrelación Trabaja. Esto implica que la entidad PROFESOR admitirá valores nulos en su clave foránea que hace referencia a DEPARTAMENTO, puesto que podrá haber profesores no asignados a ningún departamento. Pero, al contrario de lo que ocurría con las interrelaciones 1-1, aquí no se podrán evitar estos valores nulos, ya que la clave foránea debe ir necesariamente a la entidad que resulta de traducir al modelo relacional la entidad ubicada al lado N de la interrelación.

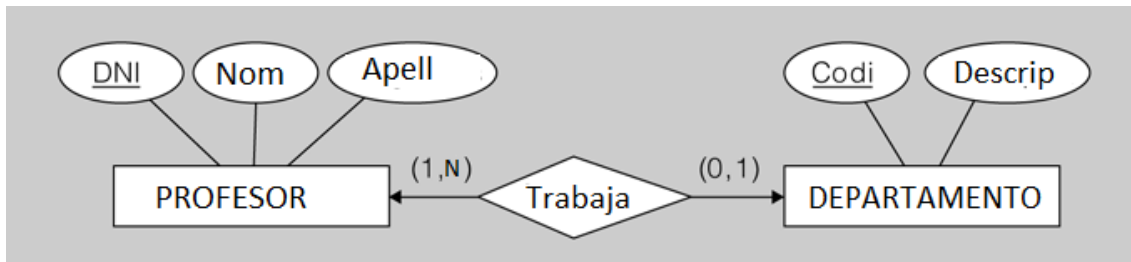


Figura 3 Interrelación binaria con conectividad 1N

c) **Conectividad M-N.** Cada interrelación M-N se transforma en una nueva relación con las siguientes características:

- Su clave primaria estará formada por los atributos de las claves primarias de ambas entidades interrelacionadas.
- Los atributos de la interrelación (si los hubiere) se convertirán en atributos de la nueva relación.

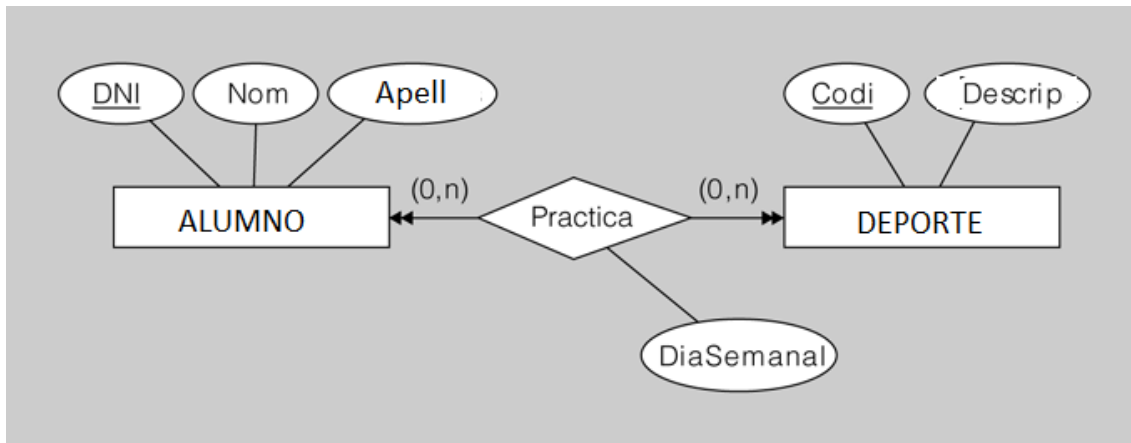


Figura 4 Interrelación binaria con conectividad M-N

Ejemplo de transformación de interrelación binaria con conectividad M-N

El diagrama ER de la figura 4 se traduce en el modelo relacional de la siguiente forma:

ALUMNO(DNI, Nombre, Apellidos)

DEPORTE(Código , Descripcion)

PRACTICA (DNIAlumno, CódigoDeporte , DiaSemanal) DONDE {DNIAlumno} REFERENCIA ALUMNO y { CódigoDeporte } REFERENCIA DEPORTE

2) **Ternárias.** Toda interrelación ternaria se transforma en una nueva relación, que tendrá por atributos los de las claves primarias de las tres entidades interrelacionadas, más los atributos propios de la interrelación, si la tuviera.

La composición de clave primaria de la nueva relación depende de la conectividad de la interrelación ternaria originaria.

a) **Conectividad M-N-P.** En este caso, la clave primaria está formada por todos los atributos que forman las claves primarias de las tres entidades interrelacionadas (si no fuera así, la clave primaria debería repetir algunas combinaciones de sus valores para modelizar todas las posibilidades, pero ésta posibilidad no está permitida dentro del modelo relacional).

Ejemplo de transformación de interrelación ternaria con conectividad MNP

El diagrama ER de la figura 5 se traduce en el modelo relacional de la siguiente forma:

ALUMNO(*DNI* , Nombre, Apellidos)

DEPORTE(*Código* , Descripción)

CURSO(*Código*)

PRACTICA(*DNIAlumno*, *CódigoDeporte*, *CodiCurso* , *DiaSemanal*) DONDE {*DNIAlumno*} REFERENCIA ALUMNO {*CódigoDeporte*} REFERENCIA DEPORTE y {*CodiCurso*} REFERENCIA CURSO

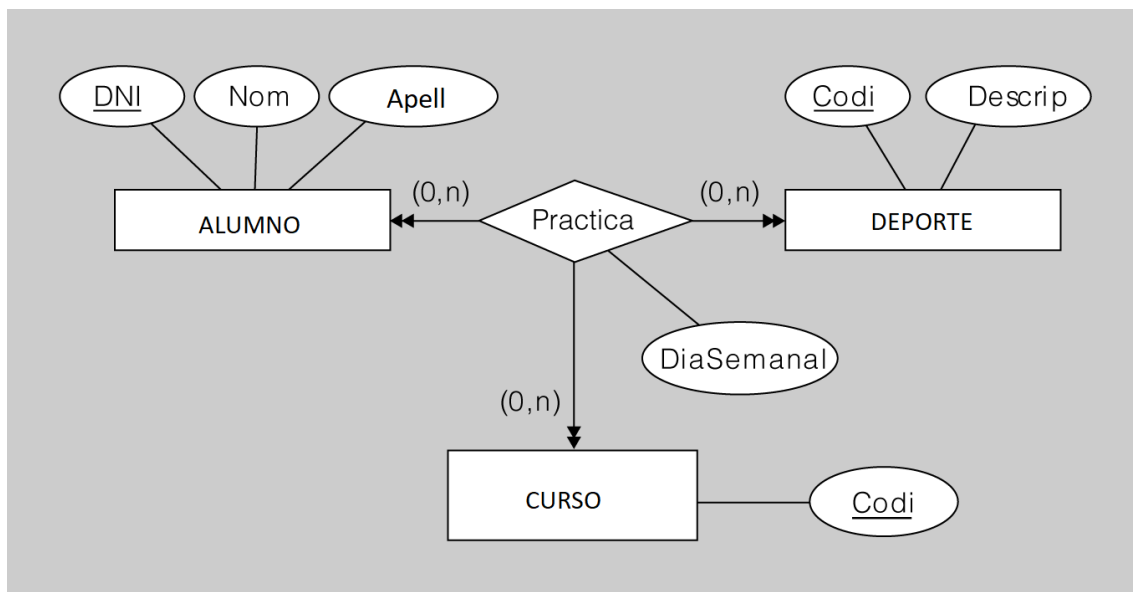


Figura 5 Interrelación ternaria con conectividad MNP

- b) **Conectividad 1-M-N.** La clave primaria está compuesta por todos los atributos que forman las claves primarias de las dos entidades que están a ambos lados de la interrelación etiquetados con una N (o con lo equivalente, una flecha de punta doble).

Ejemplo de transformación de interrelación ternaria con conectividad 1-M-N

El diagrama ER de la figura 6 se traduce en el modelo relacional de la siguiente forma:

ALUMNO(*DNI* , Nombre, Apellidos)

DEPORTE(*Código* , Descripción)

CURSO(*Código*)

PRACTICA (*DNIAlumno*, *CodiDeporte*, *CodiCurso*, *DiaSemanal*) DONDE {*DNIAlumno*} REFERENCIA ALUMNO {*CodiDeporte*} REFERENCIA DEPORTE, Y {*CodiCurso*} REFERENCIA CURSO

Fijémonos en que, en este caso, un alumno sólo puede practicar un deporte en cada curso académico y, por tanto, no es necesario incorporar la clave de la entidad DEPORTE a la clave de la relación PRÁCTICA.

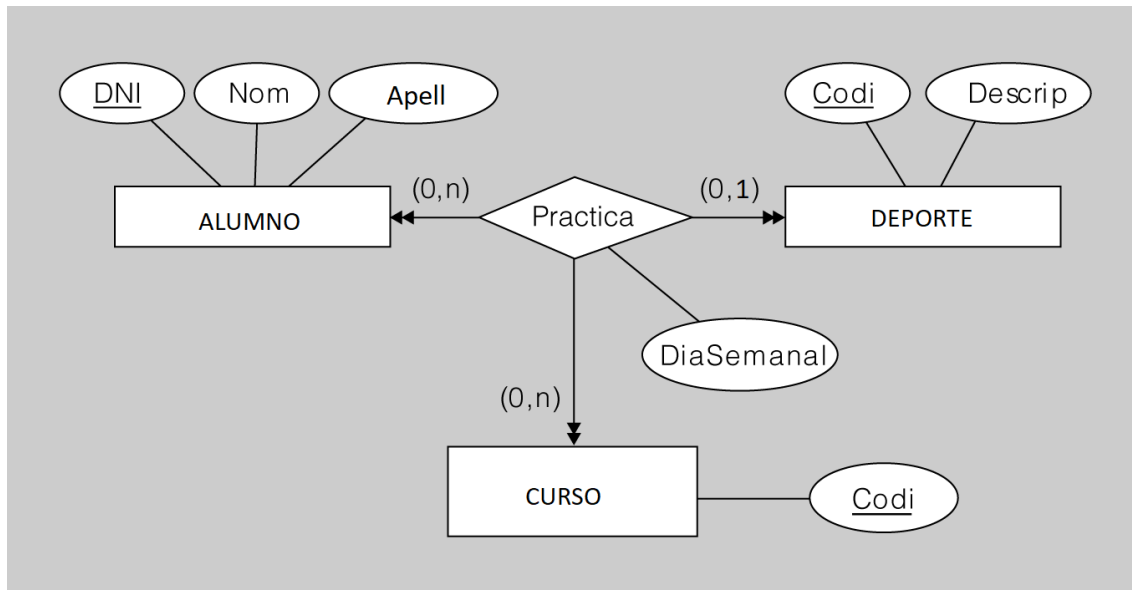


Figura 6 Interrelación ternaria con conectividad 1MN

- c) **Conectividad 1-1-N.** En estos casos, la clave primaria está compuesta por los atributos que forman la clave primaria de la entidad del lado N de la interrelación, más los atributos que forman la clave primaria de cualquiera de las otras dos entidades conectadas con cardinalidad 1.

Así pues, toda nueva relación derivada de una interrelación ternaria con conectividad 1-1-N dispondrá de dos claves candidatas. La elección de una de éstas como clave primaria de la nueva relación quedará al criterio del diseñador lógico de BD.

Ejemplo de transformación de interrelación ternaria con conectividad 1-1-N

El diagrama ER de la figura 7 puede traducirse al modelo relacional de dos formas:

ALUMNO(*DNI* , Nombre, Apellidos)

DEPORTE(*Código* , Descripción)

CURSO(*Código*)

COORDINACION (*CodiCurso*, *DNIAlumno* , *CodiDeporte*, *DiaSemanal*) DONDE
 {*DNIAlumno*} REFERENCIA ALUMNO {*CodiDeporte*} REFERENCIA DEPORTE y {*CodiCurso*}
 REFERENCIA CURSO

O bien:

ALUMNO(*DNI* , Nombre, Apellidos)

DEPORTE(*Código* , Descripción)

CURSO(*Código*)

COORDINACION (*CodiCurso*, *CodiDeporte*, *DNIAlumno*, *DiaSemanal*) DONDE
 {*DNIAlumno*} REFERENCIA ALUMNO {*CodiDeporte*} REFERENCIA DEPORTE y {*CodiCurso*}
 REFERENCIA CURSO

Puede verse que hemos modificado el nombre de la relación derivada de la interrelación para convertir el verbo originario en un sustantivo, que normalmente es más adecuado para designar relaciones.

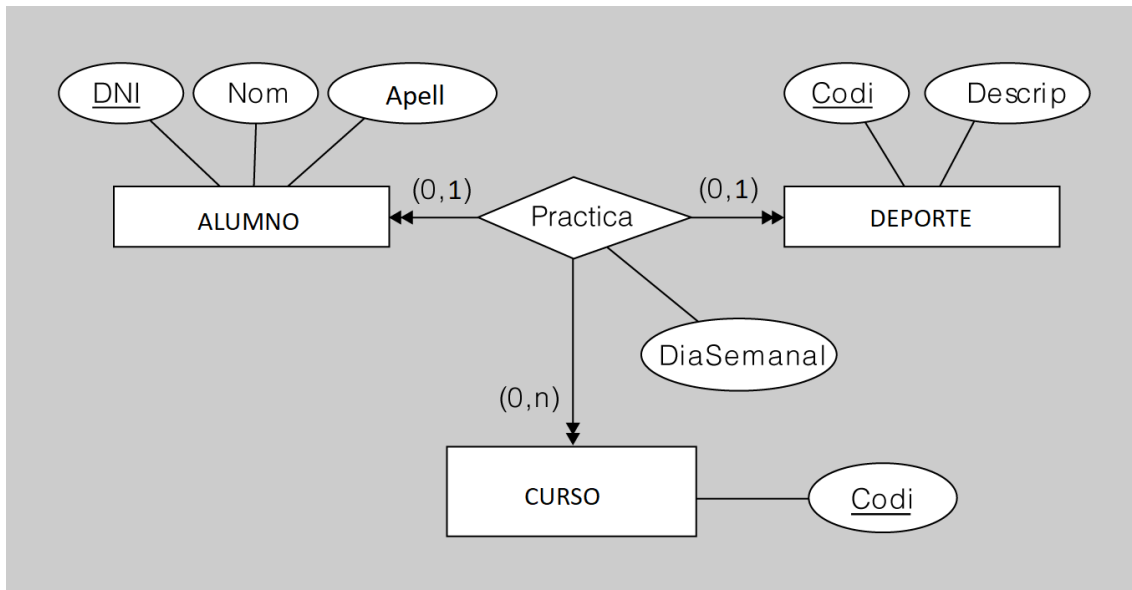


Figura 7 Interrelación ternaria con conectividad 11N

- d) **Conectividad 1-1-1.** En estos casos, la clave primaria está compuesta por los atributos que forman la clave primaria de dos entidades cualesquiera, puesto que las tres están conectadas con cardinalidad 1.

Así pues, toda nueva relación derivada de una interrelación ternaria con conectividad 1-1-1 dispondrá de tres claves candidatas. La elección de una de éstas como clave primaria de la nueva relación quedará al criterio del diseñador lógico de BD.

Ejemplo de transformación de interrelación ternaria con conectividad 1-1-1

El diagrama ER de la figura 8 puede traducirse al modelo relacional de tres formas:

ALUMNO(*DNI* , Nombre, Apellidos)

DEPORTE(*Código* , Descripción)

CURSO(*Código*)

COORDINACION (*CodiCurso*, *DNIAlumno* , *CodiDeporte*, DiaSemanal) DONDE
 {DNIAlumno} REFERENCIA ALUMNO {CodiDeporte} REFERENCIA DEPORTE y {CodiCurso}
 REFERENCIA CURSO

O bien:

ALUMNO(*DNI* , Nombre, Apellidos)

DEPORTE(*Código* , Descripción)

CURSO(*Código*)

COORDINACION (*CodiCurso*, *CodiDeporte*, *DNIAlumno*, DiaSemanal) DONDE
 {DNIAlumno} REFERENCIA ALUMNO {CodiDeporte} REFERENCIA DEPORTE y {CodiCurso}
 REFERENCIA CURSO

O bien:

ALUMNO(*DNI* , Nombre, Apellidos)

DEPORTE(*Código* , Descripción)

CURSO(Código)

COORDINACION (*CodiDeporte*, *DNIAlumno* , *CodiCurso*, *DiaSemanal*) DONDE {DNIAlumno} REFERENCIA ALUMNO {CodiDeporte} REFERENCIA DEPORTE y {CodiCurso} REFERENCIA CURSO

Fijémonos en que hemos cambiado el significado del diagrama respecto a lo que hemos representado en la figura 7: ahora un alumno sólo puede coordinar la práctica de un deporte durante un solo curso académico, a lo largo de sus estudios, para favorecer la rotación en los cargos de coordinación del centro.

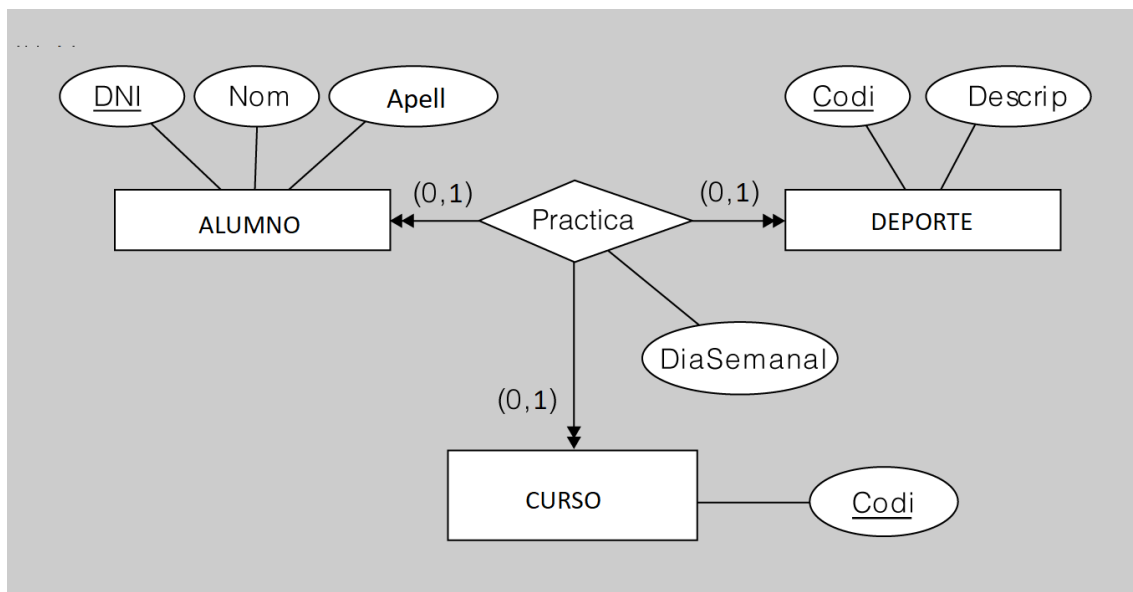


Figura 8 Interrelación ternaria con conectividad 1-1-1

- 3) **n-arias.** Cada interrelación naria se transforma en una nueva relación, que tiene como atributos las claves primarias de todas las entidades relacionadas, más los atributos propios de la interrelación originaria, si la tuviera.

La composición de la clave primaria de la nueva relación depende de la conectividad de la interrelación naria.

- a) **Conectividad de todas las entidades con cardinalidad N.** La clave primaria está formada por todos los atributos que forman las claves primarias de todas las entidades interrelacionadas (n).

Es necesario seguir el mismo mecanismo que con las interrelaciones ternarias con conectividad MNP.

- b) **Conectividad de una o más entidades con cardinalidad 1.** La clave primaria está formada por todos los atributos que forman las claves primarias de todas las entidades interrelacionadas excepto una (1-N). La entidad que no incorpora su clave primaria a la de la nueva relación debe estar forzosamente conectada con un 1.

Es necesario seguir el mismo mecanismo que con las interrelaciones ternarias con conectividad 1-M-N, 1-1-N y 1-1-1.

- 4) **Rekursivas.** Las interrelaciones recursivas traducidas se comportan de la misma forma que la del resto de interrelaciones:

- Las binarias con conectividad 1-1 y 1-N dan lugar a una llave foránea.
- Las binarias con conectividad MN y las narias originan una nueva relación.

a) **Binarias con conectividad 1-1 o 1-N.** En estas situaciones, cabe añadir a la relación surgida de la entidad originaria que se relaciona consigo misma una clave foránea que haga referencia a la propia clave primaria.

Evidentemente, los atributos de la clave foránea no pueden tener los mismos nombres que los de la clave primaria a los que se refieren, ya que ambos se encuentran en la misma relación, lo que atentaría contra los principios del modelo relacional.

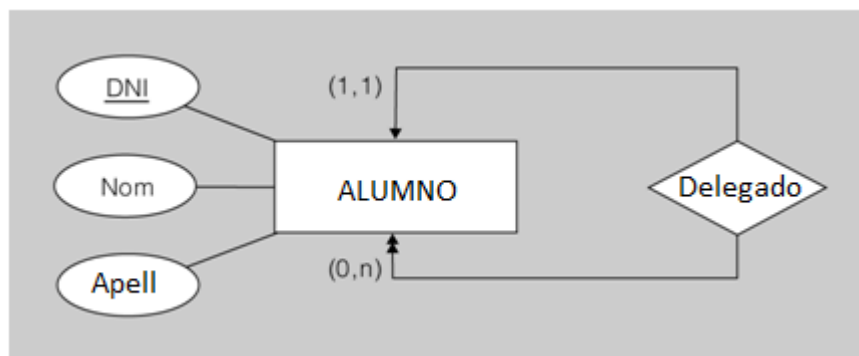


Figura 9 Interrelación recursiva binaria con conectividad 1-N

Ejemplo de transformación de interrelación recursiva binaria con conectividad 1N

El diagrama ER de la figura 9 se traduce en el modelo relacional de la siguiente forma:

ALUMNO(DNI, Nombre, Apellidos, DNIDelegado) DONDE {DNIDelegado} REFERENCIA ALUMNO

b) **Binarias con conectividad M-N.** Cuando la interrelación recursiva binaria tiene conectividad M-N se origina una nueva relación, que tiene como clave primaria los atributos que forman la clave primaria de la entidad originaria, pero dos veces, ya que es necesario modelizar el hecho de que la única entidad que interviene en la conceptualización prevista se interrelaciona consigo misma (y no con otra distinta).

Es necesario modificar convenientemente los nombres de estos atributos que están presentes dos veces en la nueva relación para que no coincidan, respetando así las directrices del modelo relacional.

Ejemplo de transformación de interrelación recursiva binaria con conectividad M-N

El diagrama ER de la figura 10 se traduce en el modelo relacional de la siguiente forma:

ASIGNATURA (Código , Descripcion)

Prerrequisito (CódigoAsignatura , CódigoPrerarequiensito) ON {CodiAsignatura} REFERENCIA ASIGNATURA (Código) y {CodiPrerrequisito} REFERENCIA ASIGNATURA (Código)

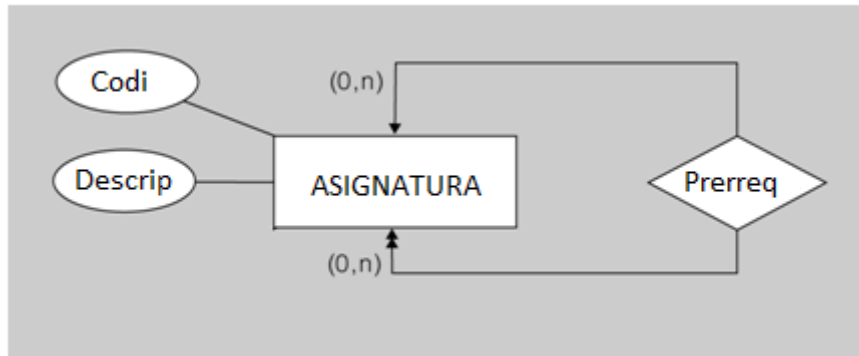


Figura 10 Interrelación recursiva binaria con conectividad M-N

- c) **n-arias**. Se origina una nueva relación, cuya clave primaria se construye de forma diferente en función de la conectividad:

Cuando la conexión de todas las entidades se produce con cardinalidad N, la clave primaria de la nueva relación se compone de todos los atributos que forman parte de las claves primarias de todas las entidades interrelacionadas (n).

Cuando la conexión de una o más de las entidades se produce con cardinalidad 1, la clave primaria de la nueva relación se compone de todos los atributos que forman las claves primarias de todas las entidades interrelacionadas salvo una (n-1). La entidad que no incorpora su clave primaria a la de la nueva relación debe estar forzosamente conectada con un 1.

Ejemplo de transformación de interrelación recursiva naria

El diagrama ER de la figura 11 se traduce en el modelo relacional de la siguiente forma:

ALUMNO(*DNI* , Nombre, Apellidos)

ASIGNATURA(*Código* , Descripcion)

DELEGADO (*DNIAlumno* , *CódigoAsignatura* , *DNIDelegado*) DONDE {DNIAlumno} REFERENCIA ALUMNO, {CodiAsignatura} REFERENCIA ASIGNATURA y {DNIDelegado} REFERENCIA ALUMNO

Fijémonos en que hemos incorporado a la clave primaria de la nueva relación los atributos que forman las claves primarias de las dos entidades conectadas con cardinalidad N, es decir, ASIGNATURA y ALUMNO, pero desde la posición de los alumnos que no son delegados.

De esta forma, se modeliza el hecho de que cada alumno tiene un delegado para cada asignatura, y que el delegado de cada asignatura representa una pluralidad de alumnos.

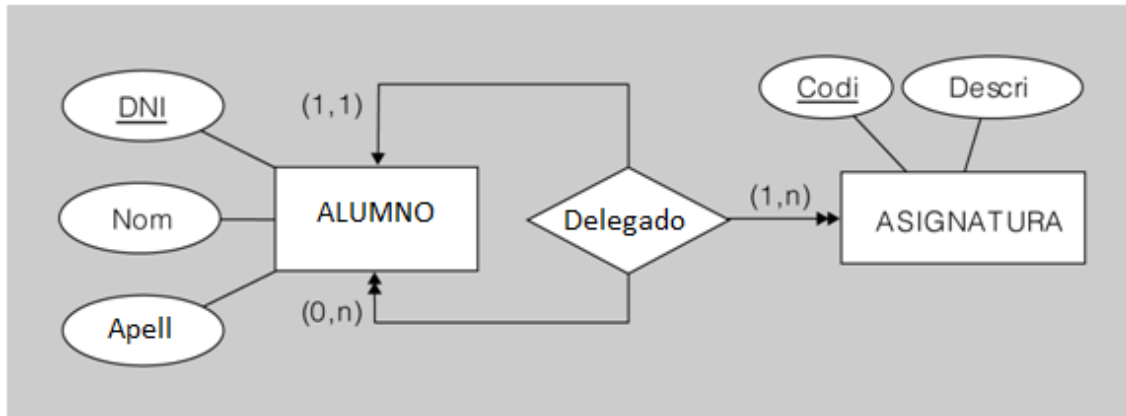


Figura 11 Interrelación recursiva naria

1.3.3. Entidades débiles

Dado que las entidades débiles siempre están situadas en el lado N de una interrelación 1-N que les sirve para completar la identificación inequívoca de sus instancias, la relación derivada de la entidad débil debe incorporar a su clave primaria los atributos que forman la clave primaria de la entidad de la que son tributarias. Dichos atributos constituyen, simultáneamente, una clave foránea que hace referencia a la entidad de la que dependen.

Ejemplo de transformación de entidad débil

El diagrama ER de la figura 12 se traduce en el modelo relacional de la siguiente forma:

CICLO(Código)

ASIGNATURA (CodiCiclo, CódigoAsignatura) ON {CodiCiclo} REFERENCIA CICLO

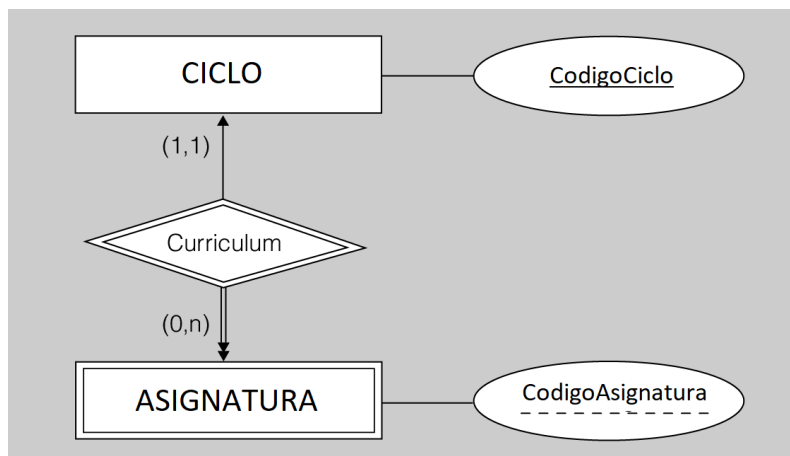


Figura 12 Entidad Débil

1.3.4. Generalización y especialización

En estos casos, tanto la entidad superclase como las entidades de tipo subclase se transforman en nuevas relaciones.

La relación derivada de la superclase hereda de ésta la clave primaria. Además, se encarga de almacenar los atributos comunes a toda su especialización o generalización.

Las relaciones derivadas de las entidades de tipo subclase también tienen, como clave primaria, la clave de la entidad superclase, que al mismo tiempo actúa como clave foránea, al referenciar a la entidad derivada de la superclase.

Ejemplo de transformación de generalización o especialización

La figura 13 muestra un encadenamiento de generalizaciones o especializaciones. Si lo traducimos a un modelo relacional obtenemos el siguiente resultado:

PERSONA(DNI , Nombre, Apellidos, Teléfono)

PROFESOR(DNI , Sueldo) DONDE {DNI} REFERENCIA PERSONA

ALUMNO(DNI) DONDE {DNI} REFERENCIA PERSONA

INFORMATICO(DNI , EspecialidadHardware, EspecialidadSoftware) DONDE {DNI} REFERENCIA PROFESOR

ADMINISTRATIVO(DNI , Titulación, Especialidad) DONDE {DNI} REFERENCIA PROFESOR

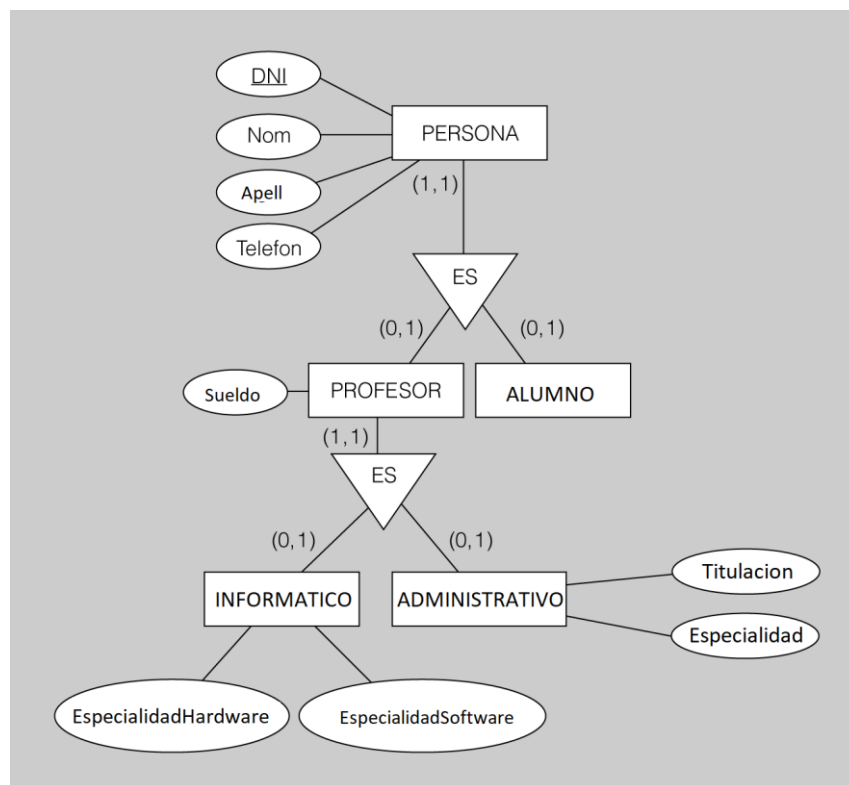


Figura 13 Generalización y especialización

1.3.5. Entidades asociativas

Las entidades asociativas se basan en una interrelación entre las entidades. La traducción de esa interrelación a un modelo relacional equivale a la traducción de la entidad asociativa.

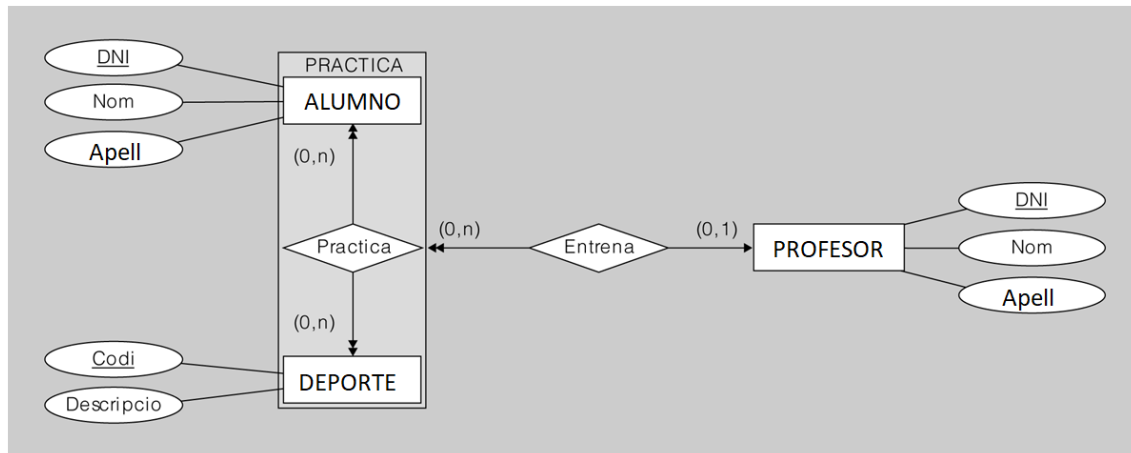


Figura 14 Entidad asociativa

Ejemplo de transformación de entidad asociativa

El diagrama ER de la figura 14 se traduce en el modelo relacional de la siguiente forma:

ALUMNO(DNI, Nombre, Apellidos)

DEPORTE(Código , Descripcion)

PROFESOR(DNI , Nombre, Apellidos)

PRACTICA (DNIALumno, CódigoDeporte , DNIProfesor) DONDE {DNIALumno} REFERENCIA ALUMNO, {CódigoDeporte} REFERENCIA DEPORTE y {DNIProfesor} REFERENCIA PROFESOR

Fijémonos como la relación PRACTICA, derivada de la entidad asociativa, incorpora una clave foránea que hace referencia a la relación PROFESOR, ya que la entidad asociativa originaria está en el lado N de una interrelación binaria con la entidad PROFESOR.