

Base de Datos

2. Interpretación de diagramas entidad/relación



Curso 2022-2023

Tabla de contenido

1. ENTIDADES Y ATRIBUTOS	4
1.1. DOMINIO DE LOS ATRIBUTOS	5
1.2. VALOR NULO DE LOS ATRIBUTOS	6
1.3. ATRIBUTOS SIMPLES Y COMPUESTOS	6
1.4. ATRIBUTOS MONOVALUADOS Y MULTIVALUADOS	7
1.5. CARDINALIDAD DE LOS ATRIBUTOS.....	8
1.6. ATRIBUTOS DERIVADOS	8
1.7. CLAVE PRIMARIA.....	9
1.8. NOTACIÓN.....	10
2. INTERRELACIONES	10
2.1. ATRIBUTOS DE LAS INTERRELACIONES	11
2.2. GRADO DE LAS INTERRELACIONES	12
2.3. CARDINALIDAD DE LAS INTERRELACIONES	13
2.3.1. <i>Interrelaciones binarias</i>	13
2.3.2. <i>Dependencias de existencia a las interrelaciones binarias</i>	15
2.3.3. <i>Interrelaciones Ternarias y n-arias</i>	16
2.3.4. <i>Límites de cardinalidad</i>	17
2.4. INTERRELACIONES RECURSIVAS.....	18
2.5. NOTACIÓN.....	20
3. ENTIDADES DÉBILES	21
3.1. NOTACIÓN.....	22
4. DIAGRAMAS ENTIDAD-RELACIÓN: DISEÑO DE BASES DE DATOS	23
4.1. FASES DEL DISEÑO DE BD	23
4.1.1. <i>Fase de diseño conceptual</i>	24
4.1.2. <i>Fase de diseño lógico</i>	25
4.1.3. <i>Fase de diseño físico</i>	25
4.2. DISEÑO CONCEPTUAL DE UNA BD.....	26
4.3. CAPTURA Y ABSTRACCIÓN DE LOS REQUERIMIENTOS DE DATOS	26
4.4. IDENTIFICACIÓN DE ENTIDADES	29
4.5. DESIGNACIÓN DE RELACIONES	30
4.6. ESTABLECIMIENTO DE CLAVES	31
4.7. ESTABLECIMIENTO DE CARDINALIDADES	32
4.8. RESTRICCIONES DE PARTICIPACIÓN Y LÍMITES DE CARDINALIDAD	33
4.9. ELABORACIÓN DE UN ESQUEMA CONCEPTUAL	34
5. EXTENSIONES DEL MODELO ENTIDAD-RELACIÓN	36
5.1. ESPECIALIZACIÓN Y GENERALIZACIÓN	36
5.2. AGREGACIONES DE ENTIDADES.....	44
6. EJEMPLO: BD DE UN INSTITUTO DE FORMACIÓN PROFESIONAL.....	47
7. EL MODELO RELACIONAL	51
7.1. ESTRUCTURACIÓN DE LOS DATOS.....	51
7.1.1. <i>Dominio</i>	52
7.1.2. <i>Esquema y extensión</i>	53
7.1.3. <i>Claves candidatas, clave primaria y claves alternativas</i>	55
7.1.4. <i>Claves foráneas</i>	56
7.1.5. <i>Operaciones con relaciones</i>	58
7.2. REGLAS DE INTEGRIDAD	59
7.2.1. <i>Unicidad de la clave primaria</i>	60
7.2.2. <i>Entidad de la clave primaria</i>	61

7.2.3.	<i>Integridad referencial</i>	61
7.2.4.	<i>Integridad del dominio</i>	65
7.3.	TRADUCCIÓN DEL MODELO ENTIDADRELACIÓN EN EL MODELO RELACIONAL.....	66
7.3.1.	<i>Entidades</i>	67
7.3.2.	<i>Interrelaciones</i>	67
7.3.3.	<i>Entidades débiles</i>	76
7.3.4.	<i>Generalización y especialización</i>	76
7.3.5.	<i>Entidades asociativas</i>	77

Las estructuras básicas del modelo Entidad-Relación (modelo ER) se corresponden, fundamentalmente, con los conceptos propuestos en la formulación original de este modelo que hizo el Dr. Peter Pin Shan Chen en su trabajo *The EntityRelationship Model Toward a Unified View of Data* 1976.

La ingeniería del software es aquella rama de la ingeniería que permite elaborar software de calidad y con un coste efectivo.

La notación de estas construcciones es fundamentalmente diagramática, aunque en algunos casos se puede añadir alguna especificación textual. Estos diagramas son generalmente conocidos como **diagramas ER** (en referencia al modelo) o **diagramas Chen** (en referencia al autor).

CASE, computer aided software ingeneering , en inglés

Los diagramas ER son muy eficaces a la hora de modelizar la realidad (empresarial o de cualquier índole) para obtener un esquema conceptual comprensible. Debido a esto, muchas de las herramientas de ingeniería del software asistida por ordenador (herramientas CASE), que también ayudan en el diseño de BD, utilizan los conceptos del modelo ER en sus diagramas.

Actualmente, tanto en la bibliografía especializada como en las herramientas CASE de diseño de BD, se pueden encontrar pequeñas variaciones a partir de la notación original propuesta inicialmente por el Dr. Chen.

La utilización de los elementos más simples del modelo ER, entidades, atributos e interrelaciones, y quizás de alguna otra construcción adicional, como las entidades débiles, pueden ser de gran utilidad en la comunicación entre los diseñadores de BD y los usuarios.

1. ENTIDADES Y ATRIBUTOS

Una **entidad** es algo que existe en el mundo real, distinguible del resto de cosas, y de la que nos interesan algunas propiedades.



Las entidades pueden tener una existencia física, como por ejemplo una persona, un coche o un libro, pero también pueden consistir en conceptos más abstractos, como un seguro o una deuda.

Ejemplo de entidad

Imaginemos que estamos diseñando la BD de un instituto de secundaria, dedicado a la enseñanza de diferentes ciclos formativos de formación profesional. Cada persona

concreta, alumna del instituto mencionado, existe en el mundo real y, por tanto, se puede considerar una entidad.

Así pues, con el término *entidad* se puede hacer referencia a un objeto específico del mundo real, pero también a un conjunto de objetos similares, de los que nos interesan las mismas características. Por lo tanto, tenemos que distinguir:

*Las **entidades** en los diagramas ER se representan con un rectángulo.*

- **Entidades instancia**, como objetos concretos del mundo real (por ejemplo, el alumno Manel Riba es una entidad instancia).
- **Entidades tipo**, como conjuntos de entidades instancia (por ejemplo, la entidad tipo alumno).

Llamamos **atributos** las características que nos interesan de las entidades.

Habitualmente, sólo nos interesará modelar una parte de los atributos de una entidad, ya que podrá haber datos que sólo serán de utilidad en ámbitos muy específicos.

Ejemplos de atributos

En una entidad instancia referente a los vehículos de un concesionario (figura 1), nos puede resultar interesante recoger ciertos datos, a fin de identificar correctamente a los coches a la hora de venderlos, como la MATRICULA, el COLOREl AÑO DE MATRICULACION, etc.

En cambio, otros datos de la misma entidad no serán de interés para nosotros, aunque sí que lo puedan ser para una BD que pertenezca a otro ámbito. Por ejemplo, desde un punto de vista sanitario, podría ser interesante registrar la altura, el peso o el grupo sanguíneo de estas mismas personas.

*Los **atributos** en los diagramas ER se representan con una elipse.*

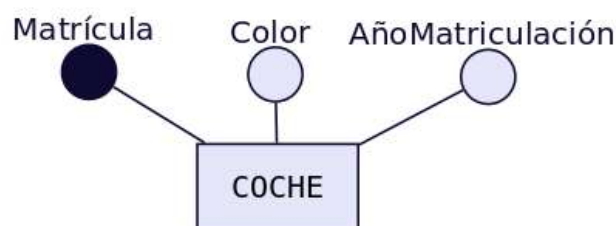


Figura 1: Ejemplos de atributos

1.1. Dominio de los atributos

Los atributos de cada entidad instancia adoptan valores concretos. Estos valores deben ser válidos.

Para que un valor de un atributo sea válido, debe pertenecer al conjunto de valores aceptables para el atributo en cuestión. Este conjunto de valores válidos denomina **dominio**.

Ejemplos de dominio y de valores válidos

El dominio del atributo COLOR de la entidad COCHE podría consistir en el conjunto de todas COLORES posibles de una longitud determinada. Serían valores válidos para el atributo COLOR, definido de este modo, "Azul", "Rojo", etc. En cambio, no lo serían, por ejemplo, una fecha de compra, un número o una cadena de caracteres que incluya alguno especial, tales como "4x4".

1.2. Valor nulo de los atributos

Los atributos de una entidad instancia pueden no tener ningún valor para algún atributo concreto. En estos casos, también se dice que el atributo tiene **valor nulo**.

Ejemplo de valor nulo

Puede suceder que un alumno no tenga teléfono. Entonces, el atributo Teléfono de la entidad ALUMNO no contendrá ningún valor o, dicho de otro modo, tendrá un valor nulo.

1.3. Atributos simples y compuestos

Se pueden considerar dos tipos diferenciados de atributos: los atributos simples y los compuestos.

Un **atributo simple** no se puede dividir en partes más pequeñas sin que ello conlleve la pérdida de su significado.

Ejemplo de atributo simple

El atributo Nombre es un atributo simple, porque su significado es indivisible (aunque en algunos casos almacene nombres compuestos, como Juan Manuel), y por tanto no tiene sentido dividir su valor en cadenas de caracteres más pequeñas por tratarse por separado.

Un **atributo compuesto** es lo que está subdividido en partes más pequeñas (que también tienen la consideración de atributos), las cuales tienen un significado propio.

Ejemplo de atributo compuesto

El atributo Apellidos se puede tratar como un atributo compuesto (figura 2), porque se puede dividir en dos partes más pequeñas (dos atributos, en definitiva) que almacenen, una, el primer apellido, y la otra el segundo apellido. Estos dos atributos se pueden tratar por separado sin problemas.

Como muchas personas extranjeras sólo tienen un apellido, en este ejemplo, el atributo Apellido1 siempre tendrá algún valor para cualquier entidad instancia, pero el atributo Apellido2 deberá admitir valores nulos.

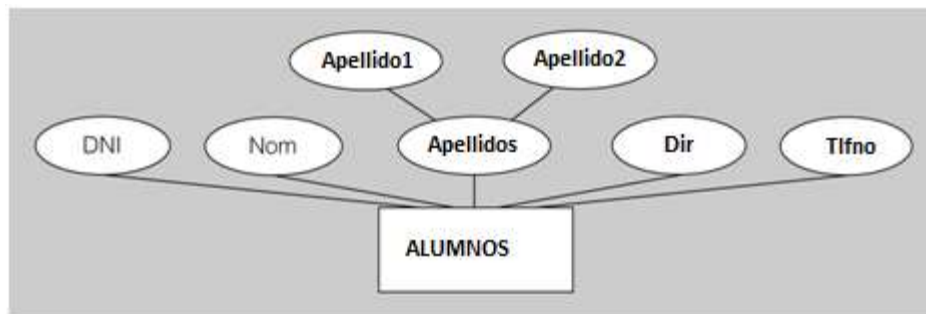


Figura 2: Ejemplo de atributo compuesto

Puede resultar interesante utilizar atributos compuestos si nos consta que los usuarios se referirán, en ocasiones, en el atributo globalmente considerado, ya veces a sus componentes por separado.

Por otra parte, los atributos compuestos agrupan los atributos relacionados, estructurándolos jerárquicamente, por lo que normalmente contribuyen a la comprensibilidad de los modelos.

1.4. Atributos monovaluados y multivaluados

En el modelo relacional ...

... los atributos resultantes sólo pueden ser simples y monovaluados. Pero el modelo ER también puede servir para hacer derivar el modelo conceptual resultante hacia otros modelos lógicos que sí acepten los atributos compuestos o los multivaluados.

Otra forma de caracterizar los atributos es en función de si son atributos monovaluats o multivaluados.

*Un atributo **monovaluado** es lo que sólo puede almacenar, como máximo, un solo valor para cada entidad instancia concreta, en un momento determinado.*

Ejemplo de atributo monovaluado

Es evidente que cada persona sólo puede tener un DNI válido. Por lo tanto, el atributo DNI de la entidad ALUMNO deberá tratarse necesariamente como un atributo monovaluado.

Un **atributo multievaluado** puede almacenar, para cada entidad instancia concreta, diferentes valores al mismo tiempo.

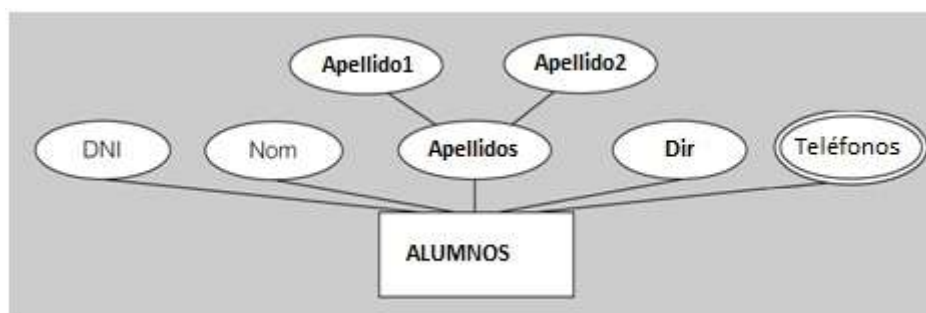


FIGURA 3: Ejemplo de atributo multivaluado

Ejemplo de atributo multivaluado

En el mundo real, una persona puede tener más de un teléfono (figura 3). Por ejemplo, puede disponer de un teléfono fijo en el domicilio particular, de otro al trabajo, y además puede tener un teléfono móvil. Por lo tanto, el atributo Teléfono de la entidad ALUMNO se puede tratar como un atributo multivaluado.

*Los atributos **multivaluados** se representan en los diagramas ER con una **elipse de doble trazo**.*

1.5. Cardinalidad de los atributos

Si es necesario, se pueden especificar a continuación del nombre del atributo, entre paréntesis y separados por comas, el límite máximo y el mínimo de valores que se han de almacenar, esto es la **cardinalidad de los atributos**. Y se pueden presentar las siguientes opciones:

- NomAtributo (1, 1): atributo univaluado obligatorio (predeterminado, si no se especifica nada).
- NomAtributo (0, 1): atributo univaluado opcional (admite valores nulos).
- NomAtributo (1, n): atributo multivaluado obligatorio (no admite valores nulos).
- NomAtributo (0, n): atributo multivaluado opcional (admite valores nulos).

Ejemplos de límites superior e inferior

Siguiendo con el caso del atributo Teléfono, se podría establecer, por ejemplo, un límite inferior a 0 (ya que un alumno puede no disponer de ningún teléfono durante un período de tiempo determinado) o 1 (si queremos obligar al alumno a dar un teléfono de contacto, aunque no sea el suyo, sino el de un familiar, amigo o vecino).

Y se podría limitar el número máximo de teléfonos almacenar, por ejemplo, a 2 (si se prevé la posibilidad, bastante habitual, tener un fijo y un móvil) o 3 (si, además, consideramos la posibilidad de registrar el teléfono del centro de trabajo).

1.6. Atributos derivados

Se dice que un **atributo** es **derivado** cuando su valor se puede calcular a partir de otros atributos o bien de otras entidades interrelacionadas.

Cuando un atributo sirve para calcular el valor de un atributo derivado, se le considera atributo base de este.

Ejemplos de atributo derivado

Podríamos necesitar saber cuál es la edad en años de los alumnos, a fin de permitirles salir o no del instituto durante los períodos de recreo, en función de aquella (figura 4). Si la entidad ALUMNO tiene un atributo llamado FechaNacimiento, podríamos modelizar otro derivado, llamado Edad, que se calculara a partir de la fecha actual (tomando la fecha del sistema) de la fecha de nacimiento (registrada en el atributo FechaNacimiento).

También podríamos necesitar saber el número total de asignaturas a las que está matriculado cada alumno. Podríamos establecer un atributo derivado llamado NombreAsignaturas, cuyo valor se cálculo es en función del número de ocurrencias de

otra entidad tipo llamada ASIGNATURA interrelacionadas con cada una de las instancias de la entidad ALUMNO.

Los **atributos derivados** en los diagramas ER se representan con una **elipse de trazo discontinuo**.



FIGURA 4: Ejemplo de atributo derivado

Los atributos derivados constituyen una redundancia, es decir, una repetición normalmente innecesaria de datos. Por este motivo, los datos de los atributos derivados incluidos en los diagramas ER no suelen almacenar (y muy especialmente si traducimos este esquema conceptual al esquema lógico más frecuentemente utilizado, es decir, al modelo relacional), sino que se calculan cuando es necesario.

1.7. Clave primaria

Una entidad instancia concreta debe poder distinguirse del resto de objetos del mundo real. Por lo tanto, cualquier modelización ER debe indicar, para toda entidad tipo, un atributo o un conjunto de atributos que la permita identificar unívocamente.

La **clave primaria** en los diagramas ER se representa **subrayando los atributos** que la forman.

El atributo o conjunto de atributos que identifican unívocamente las entidades instancia denominan **clave primaria** de la entidad.

Ejemplos de clave primaria

Podríamos seleccionar el atributo DNI de la entidad ALUMNO como clave primaria (figura 5), ya que sabemos que en el mundo real no deben existir dos documentos de identidad iguales y, por tanto, nos servirá con toda seguridad para distinguir cualquier alumno del resto.

En los países donde no existen documentos de identidad, como los anglosajones, deberíamos optar por una solución alternativa. Podríamos añadir a nuestro modelo un atributo identificador, de tipo código, aunque este no existiera en el mundo real: CodigoAlumno.

O también podríamos considerar como clave primaria un conjunto de atributos que hiciera imposible o, cuando menos, muy difícil, que se repitieran las combinaciones de sus valores para diferentes entidades instancia: Nombre + Apellidos + Teléfono.



FIGURA 5: Ejemplo de clave primaria

1.8. Notación

Notaciones ER alternativas

Actualmente no existe ninguna notación estandarizada universalmente para representar los esquemas del modelo ER. Cada recurso bibliográfico o cada software de diseño presenta, pues, variaciones y ampliaciones sobre la reducida notación propuesta originalmente por Peter Chen.

El modelo ER nos permite representar entidades y atributos mediante una sencilla notación diagramática.

En esta representación respetaremos las características siguientes:

- Como regla general, no usaremos acentos ni caracteres especiales, sólo letras y cifras.
- Representaremos las entidades tipo escribiendo su nombre en mayúsculas y en singular, dentro de un rectángulo.
- Representaremos cada atributo escribiendo su nombre con la primera letra en mayúscula y el resto en minúsculas, dentro de una elipse unida con un guión con el rectángulo que representa la entidad tipo de la que forman parte:
 - Si un atributo tiene un nombre compuesto, cada nombre comenzará con mayúscula para hacerlo más lector. Por ejemplo, **TelefonFijo**, **TelefonoMovil**.
 - Si el nombre de un atributo corresponde a unas siglas, debe ir íntegramente en mayúsculas, como DNI (documento nacional de identidad).
 - Las elipses de los atributos en que se puede descomponer un atributo deben ir unidas con un guión con la elipse del atributo compuesto.
 - La elipse de un atributo multivaluado estará formada por un trazo doble.
 - Los límites de un atributo multivaluado, en caso de existir, se especificarán a continuación del nombre del atributo, entre paréntesis y separados por una coma.
 - La elipse de un atributo derivado estará formada por un trazo punteado.
 - Los atributos que forman parte de una clave primaria deben ir subrayados.

Si hemos de establecer cualquier otra característica de los datos que no tenga predefinida una notación diagramática concreta, tendremos que añadir al diagrama las especificaciones textuales necesarias.

2. INTERRELACIONES

Una **interrelación** consiste en una asociación entre dos o más entidades.

Con el término **interrelación** podemos hacer referencia tanto a una asociación concreta entre diferentes entidades instancia, así como a una asociación de carácter

más genérico, entendida como un conjunto de asociaciones de la misma tipología, entre diferentes entidades tipo.

Ejemplo de interrelación

Ya conocemos la entidad ALUMNO (figura 6). Pero para diseñar la BD de nuestro instituto necesitaremos más entidades. Por ejemplo, será conveniente disponer de una entidad para almacenar las asignaturas que conformen la oferta formativa del centro. Podemos nombrar, esta nueva entidad, ASIGNATURA.

En un centro educativo, los alumnos se matriculan de asignaturas. Pues bien, para modelizar esta característica del mundo real, no necesitaremos ninguna nueva entidad. Sólo tendremos que establecer una asociación entre las dos entidades de que disponemos, ALUMNO y ASIGNATURA, mediante una interrelación.

De este modo, modelizar la asociación de cada alumno con todas las asignaturas en que esté matriculado, y, recíprocamente, de cada asignatura con todos los estudiantes respectivos. Podríamos llamar a esta interrelación, por ejemplo, Matricula.

Las interrelaciones en los diagramas ER se representan con un rombo.

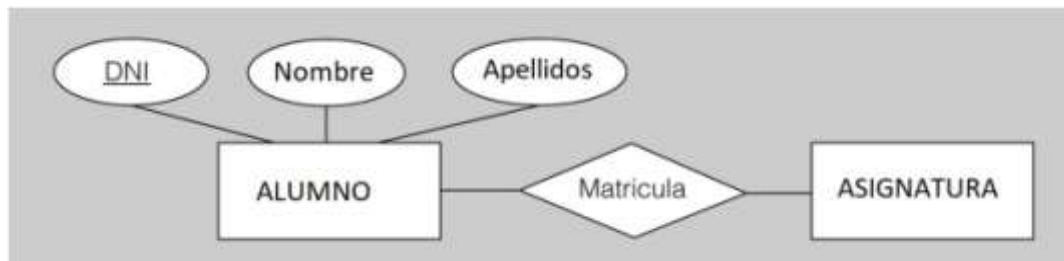


FIGURA 6: Ejemplo de interrelación

2.1. Atributos de las interrelaciones

A veces, nos puede interesar reflejar algunas características de determinadas interrelaciones. La manera de hacerlo es añadir los atributos necesarios, como haríamos si trabajáramos con entidades. Estos atributos son los **atributos de la interrelación**.

Ejemplo de atributo de interrelación

La secretaría de nuestro instituto necesitará tener constancia, como mínimo, de la nota final obtenida por cada alumno en cada asignatura en que se haya matriculado alguna vez (figura 7).

La manera más sencilla de hacerlo sería añadir, a la interrelación Matricula, un atributo llamado, por ejemplo, NotaFinal, que sirve para almacenar este dato para cada asociación existente entre instancias de las entidades ALUMNO y ASIGNATURA.

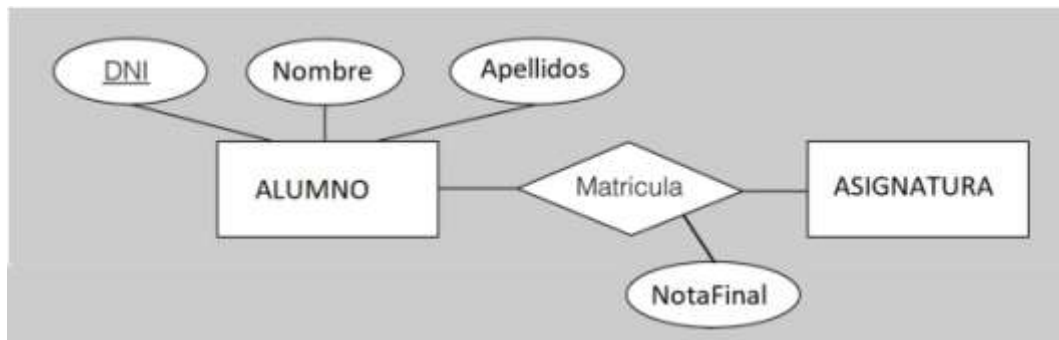


FIGURA 7: Ejemplo de atributo de interrelación

Las propiedades de los atributos de las interrelaciones son idénticas a las descritas previamente en relación a los atributos de las entidades.

2.2. Grado de las interrelaciones

El **grado de una interrelación** depende del número de entidades que esta asocia.

Encontramos un ejemplo de interrelación de **grado dos** (sólo asocia dos entidades: ALUMNO y ASIGNATURA) en la figura del subapartado "Atributos de las interrelaciones".

Las interrelaciones de grado dos también se llaman *binarias*. Y las de grado superior a dos se denominan genéricamente *n-arias*. Las interrelaciones *n-arias* de grado tres también pueden ser llamadas *ternarias*, y las de grado cuatro, *cuaternarias*.

Ejemplo de interrelación de grado tres

Hasta ahora, la interrelación Matrícula sólo permite almacenar una matrícula de cada alumno en cada asignatura, y su atributo NotaFinal sólo permite reflejar una sola nota final de curso (figura 8).

Pero este esquema no permite modelizar el hecho de que un alumno puede tener que matricularse más de una vez de una misma asignatura (y obtener una nota final en cada nueva matrícula) hasta obtener una calificación igual o superior al aprobado.

Una manera de conseguir representar esta característica del mundo real consistiría en añadir, en nuestro diseño, una nueva entidad que hiciera referencia al elemento temporal. La podríamos llamar CURSO, por ejemplo.

Y, a continuación, simplemente la interrelación Matrícula (conservando el atributo NotaFinal) interrelacione tres entidades: ALUMNO, ASIGNATURA y CURSO.

Y el nuevo esquema ya permitirá registrar matrículas sucesivas de un mismo alumno en una misma asignatura, pero a lo largo de diferentes cursos académicos, con las respectivas calificaciones obtenidas.

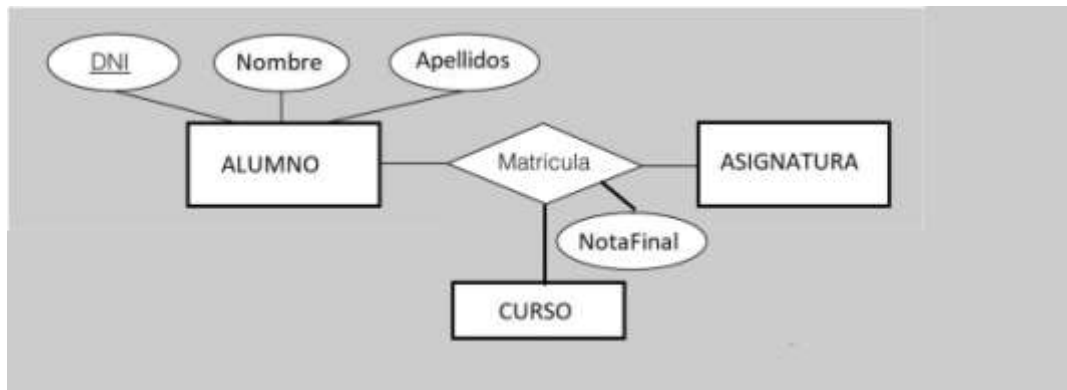


FIGURA 8: Ejemplo de interrelación de grado tres

2.3. Cardinalidad de las interrelaciones

La **cardinalidad** (también llamada *conectividad*) de una interrelación indica el tipo de correspondencia que hay entre las ocurrencias de las entidades que ella misma permite asociar.

2.3.1. Interrelaciones binarias

Tratándose de interrelaciones binarias, la cardinalidad expresa el número máximo de instancias de una de las entidades con las que una instancia de la otra entidad puede estar asociada según la interrelación en cuestión.

*Una **N** en un lado de la interrelación también se representa frecuentemente con un **asterisco** (*)*

Las interrelaciones binarias pueden ofrecer tres tipos de conectividad:

Uno a uno (1: 1)

Uno a muchos (1: N)

Muchos a muchos (N: M)

Un 1 junto a una entidad indica que, como máximo, sólo una de sus instancias (la cual podrá variar en cada caso) tendrá la posibilidad de estar asociada con cada una de las instancias de la otra entidad.

Si más de un extremo de la interrelación tiene una N, por razón de elegancia se representa con consonantes sucesivas, empezando por M: M, N, P, Q, etc.

La cardinalidad 1 también se puede representar convirtiendo la línea que une la interrelación con la entidad en una flecha que apunte hacia la entidad.

En cambio, una N (o una M) junto a una entidad indica que será una pluralidad de sus instancias (las cuales también podrán variar en cada caso) la que tendrá la posibilidad de estar asociada con cada una de las instancias de la otra entidad.

La cardinalidad N (o M) también se puede representar con una flecha de doble punta que vaya de la interrelación hacia la entidad.

Es muy importante darse cuenta de que, independientemente del tipo de conectividad, una interrelación sólo permite asociar una sola vez unas entidades instancia determinadas entre ellas.

Ejemplo de conectividad 1: 1

Como los alumnos, los profesores también forman parte de la comunidad educativa (además de otros colectivos que de momento no necesitamos tener en cuenta, ver figura 9).

Los profesores organizan profesionalmente en departamentos, en función de su especialidad (por ejemplo: matemáticas, filosofía, informática, etc.).

Para reflejar estas dos realidades, tendremos que añadir a nuestro modelo dos nuevas entidades: PROFESOR y DEPARTAMENTO.

Cada departamento es coordinado por un solo profesor, y un profesor sólo puede coordinar un solo departamento. Para reflejar esta circunstancia, deberemos establecer una interrelación entre las entidades PROFESOR y DEPARTAMENTO con cardinalidad 1: 1. Podemos nombrar la nueva interrelación Coordina.

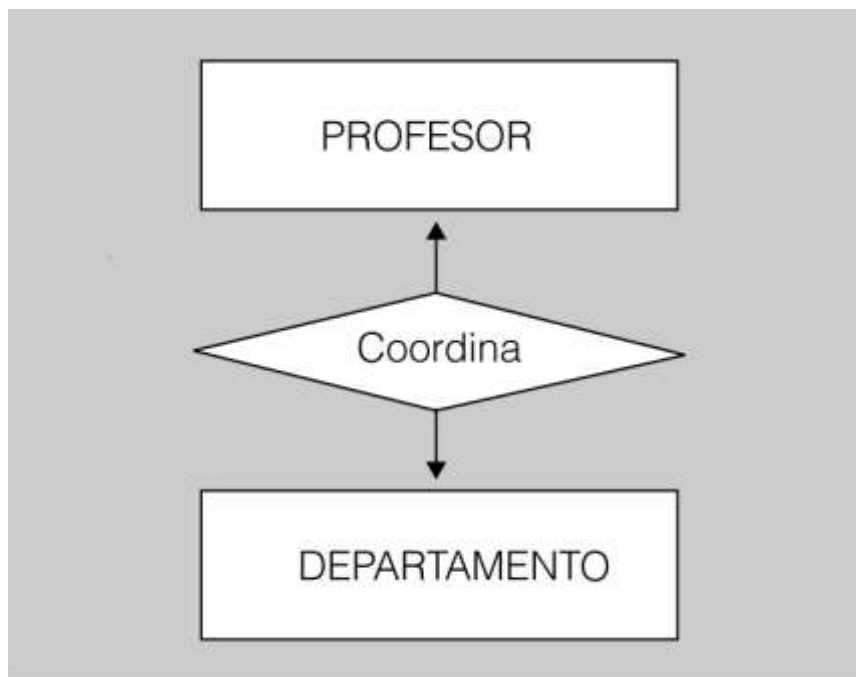


FIGURA 9: Ejemplo de conectividad 1: 1

Ejemplo de conectividad 1: N

Ya sabemos que todo profesor de instituto está asignado a un departamento (figura 10). Pero todavía nos falta establecer una nueva interrelación entre PROFESOR y DEPARTAMENTO que refleje esta realidad. La podemos llamar **TRABAJA**.

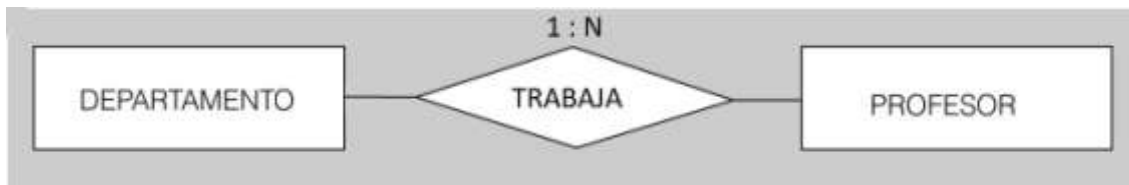


FIGURA 10: Ejemplo de conectividad 1: N

Como cada profesor sólo puede trabajar en un departamento, junto a la interrelación que conecta la entidad DEPARTAMENTO, irá un 1.

Inversamente, como que cada departamento puede tener más de un profesor asignado, junto a la interrelación que conecta la entidad PROFESOR, irá una N.

Ejemplo de conectividad N: M

Imaginemos que en nuestro instituto se organizan actividades deportivas extraescolares (figura 11). Hay que incorporar a nuestro modelo una nueva entidad (que podemos llamar DEPORTE, por ejemplo) y una nueva interrelación que la asocie con la entidad ALUMNO (que podemos llamar Practica).

Los alumnos tienen la posibilidad de inscribirse como practicantes de uno o más deportes. Y los deportes, evidentemente, pueden ser practicados por más de un alumno. Por lo tanto, a un lado de la interrelación, irá una N, y al otro una M.

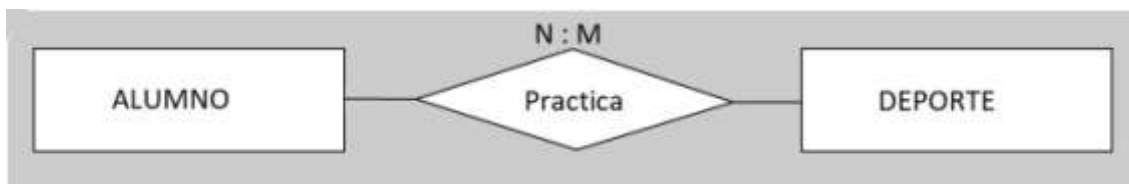


FIGURA 11: Ejemplo de conectividad M: N

2.3.2. Dependencias de existencia a las interrelaciones binarias

A veces, una entidad instancia sólo tiene sentido si existe al menos otra entidad instancia asociada con ella mediante una interrelación binaria determinada. En estos casos, se dice que la última entidad es una **entidad obligatoria** para la interrelación. De lo contrario, se dice que se trata de una **entidad opcional** para la interrelación.

Las entidades opcionales en los diagramas ER se representan superponiendo un círculo a la línea que une la entidad a la relación.

Las entidades obligatorias en los diagramas ER se representan superponiendo un pequeño guión a la línea que une la entidad a la relación.

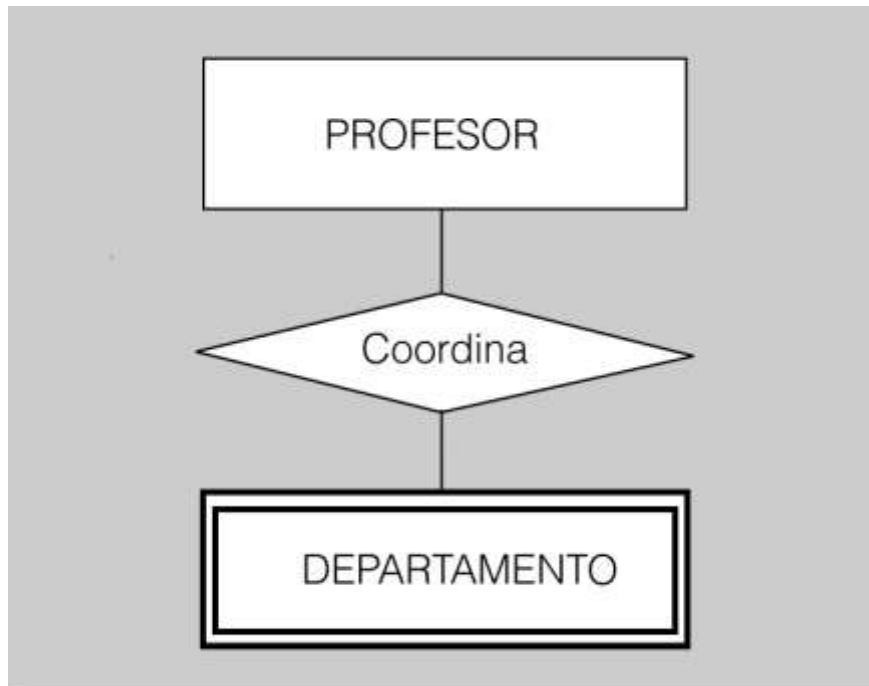


FIGURA 12: Ejemplo de dependencia de existencia

Un círculo en la línea de conexión entre una entidad y una interrelación indica que la entidad es opcional en la interrelación. La obligatoriedad de una entidad en una interrelación indica con un guión perpendicular a la línea que une la entidad con la interrelación. Si no se consigna ni un círculo ni una línea perpendicular, se considera que la dependencia de existencia es desconocida.

Tendremos en cuenta esta característica sólo por lo que haga a las interrelaciones binarias, pero no a las n-arias.

Ejemplo de dependencias de existencia

La entidad PROFESOR es obligatoria en la interrelación Coordina (figura 12). De este modo, se indica que no puede existir un departamento que no tenga ningún profesor que haga de coordinador del departamento. La entidad DEPARTAMENTO, en cambio, es opcional en la interrelación Coordina, ya que la mayoría de los profesores no coordinarán ningún departamento.

2.3.3. Interrelaciones Ternarias y n-arias

La **cardinalidad de las interrelaciones n-arias** expresa el número máximo de instancias de una de las entidades con las que una combinación concreta de instancias de las otras entidades puede estar asociada según la interrelación en cuestión.

Las interrelaciones ternarias pueden ofrecer cuatro tipos de conectividad:

- 1: 1: 1
- 1: 1: N
- 1: M: N
- M: N: P

Donde 1 indica que como máximo sólo una de sus instancias (la cual podrá variar en cada caso) tendrá la posibilidad de estar asociada con cada combinación concreta de

instancias de las otras entidades. Y en que N, M o P indica que varias instancias pueden estar relacionadas con cada combinación de instancias de las otras entidades.

En general, las interrelaciones narias pueden ofrecer $n + 1$ tipo de conectividad. Así, por ejemplo, una interrelación cuaternaria (es decir, n-aria de grado 4) tendrá cinco tipos posibles de cardinalidad (porque en este caso $n + 1 = 4 + 1 = 5$).

Ejemplo de conectividad M: N: P

Ya conocemos la interrelación Matrícula, que asocia las entidades ASIGNATURA, ALUMNO y CURSO. Pero todavía no hemos establecido sus cardinalidades (figura 13).

Un alumno, en un curso determinado, puede matricularse de varias asignaturas. Por tanto, junto a la entidad ASIGNATURA, habrá una N (pero si sólo se pudiera matricularse de una sola asignatura, debería haber un 1).

Un alumno puede tener que matricularse de una misma asignatura durante más de un curso académico, hasta que la supere. Por tanto, junto a la entidad CURSO, habrá una N (pero si sólo fuera posible matricularse una vez de una asignatura, debería haber un 1).

Y es evidente que, durante un curso académico, diferentes alumnos pueden estar matriculados en una misma asignatura. Por tanto, junto a la entidad ALUMNO, también habrá una N (pero si solo se aceptara la matrícula de un alumno por asignatura y curso, debería haber un 1).

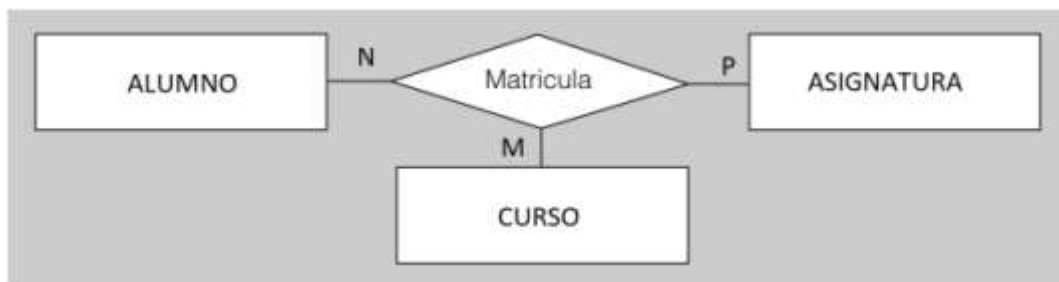


FIGURA 13: Ejemplo de conectividad M: N: P

2.3.4. Límites de cardinalidad

A veces, puede resultar útil establecer **límites mínimos y máximos a las cardinalidades de las interrelaciones** . Para ello, basta con añadir una etiqueta del tipo *mín..màx* , para expresar los límites respectivos, junto a la línea que une cada entidad con la interrelación.

Los valores *mín* y *màx* podrán tener los siguientes valores:

- Cero, para indicar la posibilidad de que no exista ninguna asociación entre instancias.
- Cualesquiera números enteros, para indicar un límite mínimo o máximo concreto de posibilidades de asociación entre instancias.
- Un asterisco (*), para indicar la posibilidad de un número ilimitado de asociaciones entre instancias.

Ejemplo de límites de cardinalidad

Ya conocemos la interrelación Coordina, que asocia las entidades PROFESOR y DEPARTAMENTO con cardinalidad 1: 1 (figura 14).

Cada departamento debe tener asignado uno, y sólo uno, profesor que el coordine. Para reflejar esta limitación, tendremos que añadir la etiqueta 1..1 junto a la línea que une la entidad PROFESOR con la interrelación Coordina.

Por otra parte, no todos los profesores se encargan de coordinar un departamento (de hecho, lo más frecuente es que no se encarguen). Y si lo hacen, sólo se pueden encargar de la coordinación de uno. Para reflejar esta limitación deberemos añadir la etiqueta 0..1 junto a la línea que une la entidad DEPARTAMENTO con la interrelación Coordina.

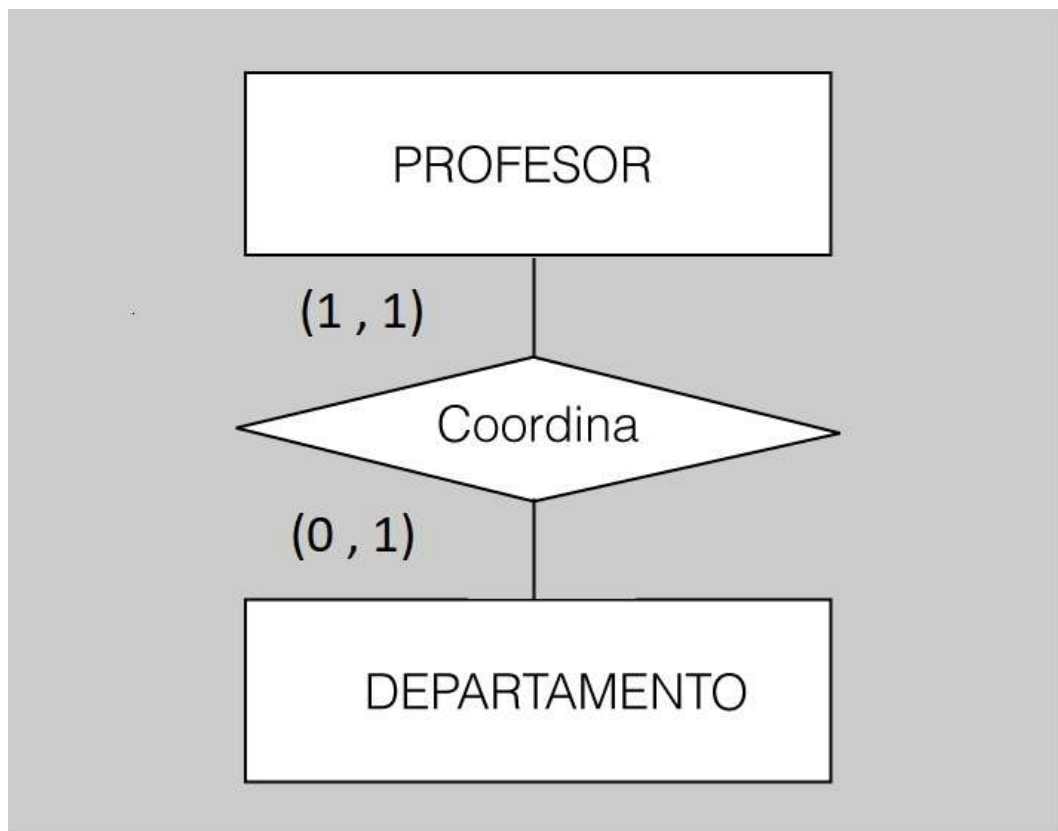


FIGURA 14: Ejemplo de límites de cardinalidad

2.4. Interrelaciones recursivas

Aunque otras interrelaciones asocian instancias de diferentes entidades, esta característica no es aplicable a las interrelaciones recursivas.

Una **interrelación recursiva** asocia las instancias de una entidad con otras instancias de la misma entidad.

Se dice que una interrelación recursiva es de grado 2 (o binaria) si sólo participa una entidad, la que se relaciona con ella misma.

Ejemplo de interrelación recursiva binaria

Imaginemos que en nuestro instituto se establece, como requisito para cursar ciertas asignaturas, el hecho de haber superado previamente otra u otras asignaturas (figura 15).

Podríamos modelizar esta situación mediante una interrelación recursiva binaria sobre la entidad PERSONA, y llamarla, por ejemplo, Supervisar.

Si consideramos que cada persona puede supervisar a una o más persona, y que al mismo tiempo cada persona es supervisada por una y solo una persona la cardinalidad debería ser 1: N.

*Las **interrelaciones recursivas** en los diagramas ER se representan **conectando una misma entidad más de una vez**, mediante una única relación.*

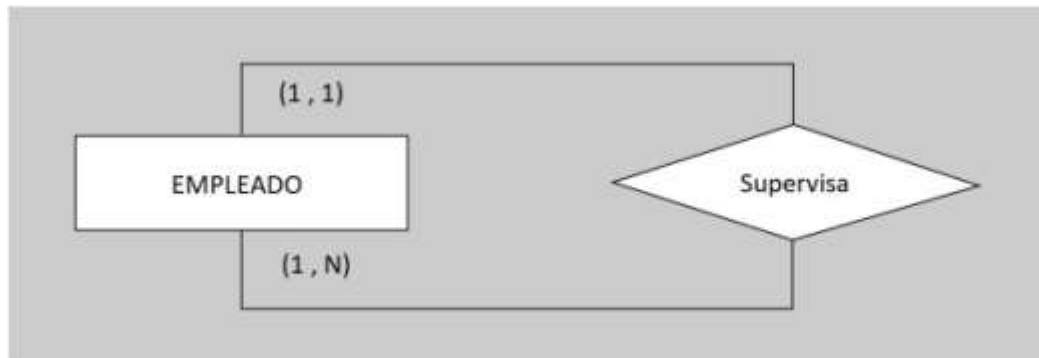


FIGURA 15: Ejemplo de interrelación recursiva binaria

Si en una interrelación recursiva participan, adicionalmente, más entidades, hablaremos de interrelaciones recursivas de grado 3 (o ternarias), de grado 4 (o cuaternarias), y así sucesivamente.

Ejemplo de interrelación recursiva ternaria

Cada alumno de nuestro instituto tiene un delegado por asignatura, que lo representa ante el profesorado que la imparte, a fin de hacer más fluidas las comunicaciones sobre las cuestiones relativas al funcionamiento de aquella que no sean de índole personal (figura 16) .

Podríamos modelizar esta situación mediante una interrelación recursiva ternaria, llamada, por ejemplo, Delegado. Una ocurrencia de esta interrelación asociará un alumno que actuará como delegado en el ámbito de una asignatura, otro alumno que actuará como estudiante de la misma asignatura, y la asignatura en cuestión.

La conectividad es 1: M: N. En ambos lados de la entidad ALUMNO hay un 1 y una N, porque, por un lado, un delegado de una asignatura puede representar más de un estudiante (N), y, por otro lado, un estudiante de una asignatura sólo puede tener un solo representante en el ámbito de esta (1). Y junto a la entidad ASIGNATURA hay una M, para que un alumno puede actuar como representante de otro en diferentes ámbitos, correspondientes a diferentes asignaturas.

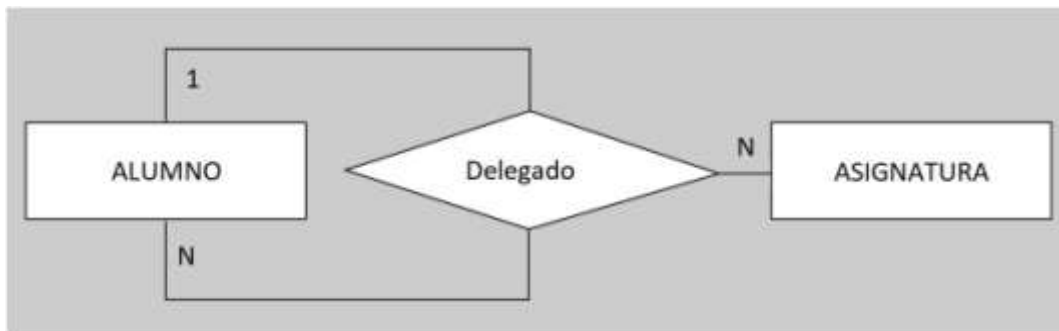


FIGURA 16: Ejemplo de interrelación recursiva ternaria

En una interrelación no recursiva, el papel, o rol, que interpreta cada entidad implicada se sobreentiende y, por tanto, no es necesario especificarlo.

En el caso de las interrelaciones recursivas, puede tener importancia especificar los diferentes papeles o **roles** que interpretan las instancias de una misma entidad, si estos roles no coinciden plenamente. Si el rol es exactamente lo mismo, no es necesario especificarlo.

Ejemplo de diferenciación de roles

Podríamos etiquetar las dos líneas de la interrelación Prerequisito como "pre" y "post", por ejemplo, a fin de modelizar en el primer caso el rol de prerequisito académico y, en el segundo caso, el rol de asignatura autorizada (figura 17).

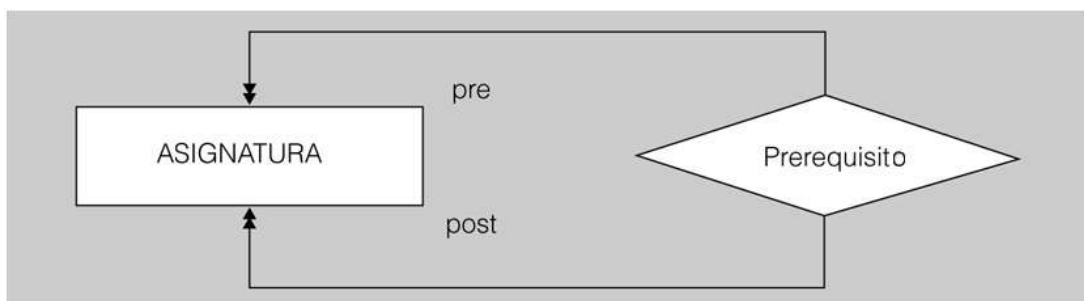


FIGURA 17: Ejemplo de diferenciación de roles

2.5. Notación

Como con las entidades, la notación diagramática para representar las interrelaciones y sus propiedades también es bastante sencilla:

- Toda interrelación se representa con un rombo, que va unido, mediante líneas, a todas las entidades que asocia.
- Los atributos de una interrelación, cuando existen, se representan de la misma manera que los atributos de una entidad.
- La conectividad de una interrelación se representa añadiendo una etiqueta con un 1 o un N, según sea necesario, a cada una de las líneas que la une con las entidades que participan.

- La opcionalidad se representa superponiendo un círculo a la línea de conexión correspondiente, y la obligatoriedad, superponiendo un pequeño guión perpendicular a la línea de conexión de que se trate.
- Si hay que establecer límites (0, entero, *) en la cardinalidad de una interrelación, debe añadirse a cada una de sus líneas de conexión una etiqueta con el límite inferior y el superior separados por dos puntos seguidos.
- La recursividad de una interrelación se representa haciendo llegar dos líneas de conexión a la misma entidad. Si participan más entidades de la misma interrelación recursiva, se harán llegar las líneas de conexión correspondientes desde la interrelación.
- Si hay que hacer una diferenciación de los roles de una interrelación recursiva, se debe añadir una etiqueta, con la especificación textual adecuada, junto a cada una de las líneas de conexión.

3. Entidades Débiles

Las entidades que disponen de un atributo o, si no, de un conjunto de atributos capaces de establecer una clave primaria que sirva para distinguir cada instancia de la entidad del resto de ocurrencias se pueden llamar, más específicamente, *entidades fuertes*.

Las **entidades débiles** son aquellas que no disponen de suficientes atributos para designar unívocamente sus instancias. Para conseguirlo, deben estar asociadas, mediante una interrelación, con una entidad fuerte que las ayude.

La interrelación entre una entidad débil y su fuerte asociada es siempre de cardinalidad 1: N, y se resta del 1 al lado de la entidad fuerte, y la N junto a la débil.

Cada instancia de una entidad débil está asociada con una única ocurrencia de la entidad fuerte (por eso es en el lado 1 de la interrelación), y así es posible completar la identificación de manera única.

Por otra parte, la entidad del lado 1 debe ser obligatoria en la interrelación que, si no fuera así, alguna instancia de la entidad débil podría no estar asociada con ninguna de las ocurrencias de la entidad fuerte y, entonces, no se podría identificar completamente.

Las entidades débiles, pues, no tienen clave primaria, pero sí un atributo (o un conjunto de atributos) llamado **discriminante**, que permite distinguir entre ellas todas las instancias de la entidad débil que dependen de una misma instancia de la entidad fuerte.

Aunque no es un caso muy frecuente, se pueden encadenar entidades débiles, de tal manera que una entidad que actúe como parte débil en la interrelación que mantenga con otra entidad, puede actuar al mismo tiempo como entidad fuerte respecto a otra entidad que, a su vez, la necesite para identificar completamente sus instancias.

Adicionalmente a la interrelación que les sirve para identificarse completamente, las entidades débiles pueden participar en otras interrelaciones, como cualquier otra entidad.

Ejemplo de entidad débil

Ha llegado el momento de establecer una clave primaria para la entidad ASIGNATURA (figura 18). Podríamos adoptar una codificación derivada de la utilizada en los currículos oficiales: C1, C2, C3, etc. (de Crédito 1, Crédito 2, y así sucesivamente). Podríamos llamar a este atributo CodiAssignatura. Pero esto no permitiría distinguir las asignaturas de los diferentes ciclos formativos impartidos en nuestro instituto.

Para lograr la identificación inequívoca de cada crédito, en primer lugar deberíamos contar con una nueva entidad llamada, por ejemplo, CICLO, para almacenar todos los ciclos impartidos en el centro. Esta entidad sería fuerte, y sus instancias se distinguirían inequívocamente las unas de las otras mediante una clave primaria que se podría decir CodiCiclo.

A continuación, deberíamos establecer una interrelación binaria llamada, por ejemplo, Curriculum, en la que participara la entidad ASIGNATURA como entidad débil, en el lado N de la interrelación, y la entidad CICLO como entidad fuerte, en el lado 1.

Las relaciones débiles en los diagramas ER se representan con un rectángulo de doble línea.

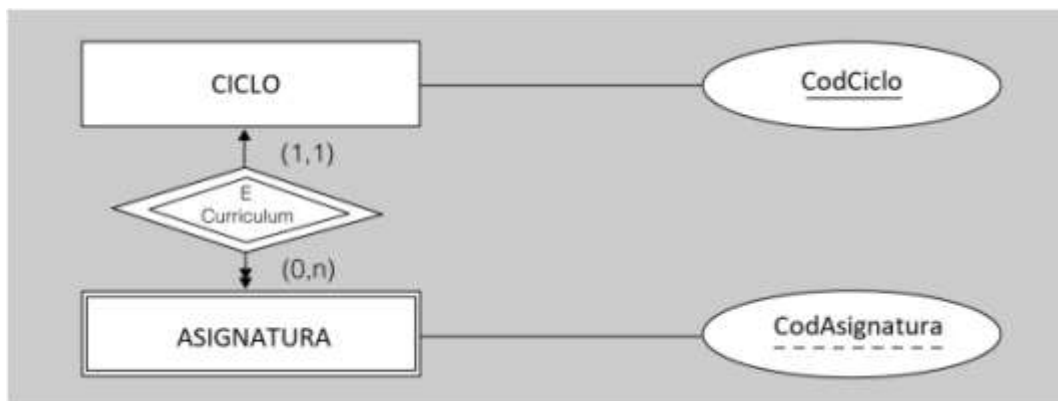


FIGURA 18: Ejemplo de entidad débil

3.1. Notación

Para incorporar las entidades débiles a los diagramas ER, hay que aplicar unas pocas reglas de notación adicionales:

- Las entidades débiles se representan escribiendo su nombre en mayúsculas y en singular, dentro de un rectángulo dibujado con una línea doble.
- La interrelación que une la entidad débil con su fuerte se representa con un rombo también de línea doble.
- El atributo o conjunto de atributos que actúen como discriminantes deben ir subrayados con una línea discontinua.

4. DIAGRAMAS ENTIDAD-RELACIÓN: DISEÑO DE BASES DE DATOS

Los diagramas Entidad-Relación son un estándar actual en el diseño de bases de datos. De hecho, se trata de una herramienta sin la que, posiblemente, las bases de datos tal como las entendemos actualmente no existirían.

Los diagramas Entidad-Relación son herramientas gráficas clave en el diseño de bases de datos. Su confección debe ser sistemática y rigurosa si se quiere obtener un sistema válido y eficiente, ya que será a partir de estos diagramas que se desarrollará toda la implementación de bases de datos en los sistemas gestores de bases de datos concretos que corresponda a cada empresa u organismo.

El diseño de BD estructura, fundamentalmente, en tres grandes etapas:

- diseño conceptual
- diseño lógico
- diseño físico

Aunque nos centramos en el estudio y en la práctica del diseño conceptual, no se deben perder de vista las otras fases del diseño, que también son importantes, a fin de obtener una visión de conjunto de todos estos procesos.

Conocer cómo se estructura el modelo Entidad-Relación es importante. Y también lo son cuestiones que nos deben permitir aprovechar la tecnología proporcionada por las BD y los correspondientes sistemas gestores, como los siguientes:

- ¿Qué entidades debe incluir una BD determinada.
- Qué relaciones se deben considerar.
- ¿Qué atributos deben existir y en qué entidades o relaciones se incorporarán.
- ¿Qué claves primarias ya se pueden establecer en la fase de diseño conceptual.

4.1. Fases del diseño de BD

Diseñar BD no es una tarea sencilla. Aunque la porción del mundo real que se quiera modelizar en un caso concreto sea relativamente pequeña, las estructuras de datos resultantes pueden llegar a tener un cierto grado de complejidad. Sin embargo, a medida que aumenta la información a considerar, y su complejidad, el modelo de datos necesario para representarla puede convertirse, ciertamente, en una construcción complicada.

Mundo Real

Cuando hablamos de *mundo real*, nos referimos en el escenario o situación concreta que se quiere modelizar para ser explotada mediante una BD.

Querer resolver de golpe toda la problemática que puede conllevar la modelización de una BD no es, pues, una opción muy realista. Al afrontar una tarea de esta envergadura, es preferible dividirla en subtarefas para simplificarla.

Así pues, resulta conveniente descomponer el diseño de BD en diferentes etapas, de modo que en cada una sólo se examinen ciertos aspectos, o tipo de problemas, para minimizar la posibilidad de error. Entonces, a partir del resultado obtenido en cada fase, se puede continuar trabajando en la fase siguiente, hasta llegar al resultado esperado, al final de la última fase.

Es habitual estructurar el diseño de BD en las tres etapas o fases:

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

4.1.1. Fase de diseño conceptual

Lo primero que hay que hacer, durante la fase de diseño conceptual, es recopilar toda la información necesaria de la parte del mundo real que nos proponemos modelizar con una BD.

Esta recopilación de información se realizará por diferentes vías, como éstas:

- Entrevistas con los futuros usuarios de la BD que se está diseñando.
- Examen de la documentación proporcionada por estos mismos usuarios.
- Observación directa de los procesos a informatizar.

A continuación, se deben estructurar convenientemente los datos necesarios para dar respuesta a todas las necesidades derivadas del conjunto de informaciones compendiadas.

El objetivo del diseño conceptual consiste en la obtención de una especificación sistemática.

Dicha especificación sistemática resultado del diseño conceptual debe cumplir dos tipos de requisitos:

- **De datos.** El modelo resultante debe tener en cuenta la estructura completa de los datos y su integridad.
- **Funcionales.** Un buen esquema conceptual también deberá prever las necesidades básicas en materia de manipulación de datos (es decir, las operaciones de inserción, borrado, consulta y modificación, de éstas). Durante las fases posteriores, puede ser conveniente depurar el diseño para optimizar las operaciones a realizar sobre los datos.

*El modelo **Entidad-Relación** (o modelo Entidad-Interrelación) también se conoce de manera abreviada como **modelo ER** (siglas correspondientes a **entity relation ship**).*

Finalmente, hay que elegir un modelo de datos de alto nivel y traducir los requisitos anteriores a un esquema conceptual de la futura BD expresado con los conceptos y la notación correspondientes. Uno de los modelos de datos de alto nivel más utilizados es el **modelo Entidad-Relación**.

Expresado en la terminología del modelo ER, el esquema de datos desarrollado durante la fase de diseño conceptual debe especificar todas las entidades necesarias, y las relaciones entre ellas, con las cardinalidades adecuadas, y también los atributos que correspondan en cada caso.

El modelo ER

Una entidad, en el modelo ER, es una abstracción que nos interesa modelizar, y mediante la cual se agrupan las instancias del mundo real que tienen unas características comunes.

Una Relación, en el modelo ER, es la asociación entre instancias de diferentes entidades tipo. Esta asociación se puede dar con varias cardinalidades.

Un atributo, en el modelo ER, es una característica de una entidad que nos interesa tener registrada.

El modelo resultante debe revisarse para garantizar la satisfacción de todas las necesidades detectadas, por un lado, y para evitar redundancias de los datos (es decir, repeticiones indeseadas de estas), de otra.

El resultado de la fase de diseño conceptual pertenece al llamado **mundo de las concepciones**, pero todavía no al **mundo de las representaciones**, ya que no se especifica ninguna representación informática concreta.

Como se puede ver, durante la fase de diseño conceptual no hay que tener en cuenta, aunque, ni el tipo de BD que se utilizará posteriormente ni, mucho menos, el SGBD o el lenguaje concreto con el que se implementará.

4.1.2. Fase de diseño lógico

En la fase de **diseño lógico**, se trabaja con el modelo abstracto de datos obtenido al final de la etapa de diseño conceptual, para traducirlo al modelo de datos utilizado por el sistema gestor de bases de datos (SGBD) con el que se quiere implementar y mantener la BD.

*Las **claves primarias** sirven para distinguir entre sí las diferentes tuplas de atributos dentro de una misma relación.*

Por tanto, a partir de esta fase de diseño, sí que hay que tener en cuenta la tecnología concreta que se debe emplear en la creación de la BD, ya que la BD resultante se puede adecuar a diferentes modelos lógicos, tales como los siguientes:

- jerárquico
- relacional
- distribuido
- Orientado a objetos

A pesar de la diversidad de posibilidades, lo cierto es que lo más frecuente, a la hora de diseñar una BD, aunque consiste en expresar el esquema conceptual en un modelo ER y, a continuación, traducirlo a un modelo relacional.

*Las **claves foráneas** son unos instrumentos destinados a permitir la Relación de la respectiva relación con otros.*

Cuando el producto de una fase de diseño lógico es una BD relacional, esta consiste en un conjunto de relaciones (de lo contrario, llamadas *representaciones tabulares*) compuestas por atributos, algunos de los cuales forman parte de claves primarias o de claves foráneas.

Resulta evidente, pues, que el resultado de la fase de diseño lógico ya se sitúa dentro del llamado **mundo de las representaciones informáticas**.

4.1.3. Fase de diseño físico

El **diseño físico** consiste en hacer ciertos tipos de modificaciones sobre el esquema lógico obtenido en la fase anterior de diseño lógico, a fin de incrementar la eficiencia.

La eficiencia de un esquema puede conllevar la modificación de algunas operaciones que deban hacerse con los datos, aunque conlleven un cierto grado de redundancia de éstas, como por ejemplo:

- Añadir algún atributo calculable en alguna relación.
- Dividir una relación en otras dos o en más.
- Incluir en la BD una relación que sea el producto de combinar dos o más relaciones.

Pero la fase de diseño físico también se caracteriza por la posibilidad de adoptar otras decisiones, relacionadas con aspectos de implementación física a más bajo nivel, y estrechamente vinculadas con el SGBD con el que se trabaja en cada caso, como los siguientes:

- Definición de índices.
- Asignación del espacio inicial para las tablas, y previsión de su crecimiento ulterior.
- Selección del tamaño de las memorias intermedias.
- Parametrización del SGBD según las opciones que éste ofrezca.

La volatilidad de los datos tiene que ver con el volumen de inserciones y borrados de las mismas.

Para tomar acertadamente estos tipos de decisiones, hay que tener en cuenta las características de los procesos que operan con los datos, la frecuencia de ejecución de los diferentes tipos de consulta, el grado de volatilidad de los datos, los volúmenes de información a almacenar, etc.

4.2. Diseño conceptual de una BD

Podemos considerar cualquier empresa, organización, institución, etc. como un sistema con reglas propias de funcionamiento. Este sistema, susceptible de ser informatizado, está compuesto por tres subsistemas:

- **Subsistema de producción.** Se encarga de realizar las actividades propias de la organización de que se trate en cada caso (por ejemplo, fabricar coches, o repararlos, o venderlos, etc.).
- **Subsistema de decisión.** Se encarga de dirigir, coordinar y planificar las actividades realizadas dentro del ámbito del subsistema de producción.
- **Subsistema de información.** Se encarga de recoger, almacenar, procesar y distribuir, todas las informaciones necesarias para el buen funcionamiento de los otros dos subsistemas.

Las BD sirven para almacenar las representaciones de las informaciones utilizadas dentro del ámbito de los sistemas de información.

Los elementos que conforman un **sistema de información** son de dos tipos:

- Datos: representaciones de las informaciones.
- Procesos: acciones ejercidas sobre los datos (consultas, modificaciones, cálculos, etc.).

En función de las observaciones anteriores, podemos afirmar que hay dos premisas que todo diseñador de BD debería tener bien asumidas antes de empezar a trabajar en cualquier proyecto:

- No es competencia del diseñador de BD, como tal, tomar decisiones sobre la porción del mundo real que quiere modelizar.
- En principio, el diseñador de BD tampoco hay que inventar características de la realidad modelizar: simplemente las debe reflejar de la manera más fiel posible en el modelo resultante.

Podemos subdividir esta etapa de diseño conceptual en dos fases sucesivas, las cuales conllevan tareas de diferentes tipos: la **recogida y abstracción de las necesidades** de la organización, por un lado, y la **elaboración de un esquema concepatual** mediante un modelo de datos concreto, de la otra.

La **recogida y abstracción de las necesidades** de la organización se lleva a cabo mediante diferentes procedimientos (entrevistas, examen de documentación, etc.), y en esta fase hay que recoger toda la información necesaria para cubrir todos los requerimientos de datos. Pero, con esta información, se debe seguir un proceso de abstracción que nos permita estructurarla y diferenciar entre las cuestiones esenciales y las accesorias.

La **elaboración de un esquema concepatual** mediante un modelo de datos concreto conlleva que toda la información recogida y debidamente estructurada se expresará en una notación estandarizada (como los diagramas ER).

4.3. Captura y abstracción de los requerimientos de datos

Para realizar la captura y abstracción de los requerimientos, en primer lugar, es necesario averiguar qué necesidades tienen los usuarios de la futura BD. A menudo, estos usuarios potenciales sólo tienen una percepción muy general de lo que necesitan.

Usuarios de las BD

Lo son tanto los operadores que interactúan habitualmente con el sistema, como los destinatarios finales de la información que se ha de extraer o que debe incluirse en la BD.

El diseñador debe saber seleccionar las cuestiones esenciales, diferenciarlas de los aspectos accesorios, y descubrir cuáles son los verdaderos intereses de los que han encargado el diseño de la BD.

El diseñador de BD es el profesional informático que se encarga de realizar las tareas que implica el diseño de las BD.

Ahora bien, el diseñador de BD, como tal, no debe decidir nada. Su tarea consiste más bien en ayudar a los usuarios potenciales a descubrir qué necesitan exactamente. Es un trabajo pesado y complicada, que sirve para descubrir el verdadero flujo de datos que se reflejará en el modelo conceptual resultante. Normalmente, esta función conlleva lo siguiente:

- Muchas entrevistas con los futuros usuarios de todos los niveles y secciones de la organización a informatizar. Hay que anotar todos los detalles surgidos durante las entrevistas, susceptibles de implementación informática.
- Examen exhaustivo de la documentación proporcionada por el cliente, con el fin de conocer el funcionamiento interno de la organización.
- Observaciones directas de los diferentes procesos de la organización.

Un buen diseñador de BD debe llegar, como mínimo, a una solución plausible para cada problema que se le haya planteado. A veces, también podrá tomar en consideración soluciones parciales alternativas. Y, en todo caso, finalmente, debe presentar un análisis completo de los requerimientos en la que concrete las especificaciones de la organización que le ha encargado el proyecto de diseño.

Un ejemplo concreto: BD de la Red de Bibliotecas de la Junta de Andalucía (ReBJA)

El ReBJA como tal no existe. Este ejemplo sólo pretende simular la metodología de trabajo habitual a la hora de diseñar conceptualmente una base de datos.

Examinar los requisitos de diseño que nos permitan establecer un modelo conceptual para una futura BD que dé servicio conjuntamente a las bibliotecas de todos los institutos de Andalucía (mediante una aplicación web ulterior, por ejemplo), a fin de poner al alcance de cualquier miembro de la comunidad educativa de secundaria la totalidad de los fondos bibliográficos y de otros recursos audiovisuales, aunque no se encuentren físicamente en el centro docente donde trabaje o estudie el usuario que solicite el préstamo, nos proveerá de un buen escenario para desarrollar un ejemplo de diseño de BD concreto.

No se trata ahora tanto de profundizar en cuestiones de detalle, como del hecho de captar globalmente como se trabaja durante la fase de diseño conceptual, y aplicar los conocimientos adquiridos previamente los cuales son necesarios para elaborar los esquemas conceptuales, en nuestro caso fundamentalmente con ayuda del modelo de datos ER.

Informe de análisis y diseño de BD

La información recogida por los diseñadores de BD después de la captura y la abstracción de los requerimientos de los datos se recoge en un informe.

Después de entrevistarnos con el alto cargo del Departamento de Educación de la Junta de Andalucía que promueve la coordinación y colaboración de las bibliotecas de todos los institutos (LOCAL) de Andalucía, con varios directores de diferentes bibliotecas interesadas en adherirse a la futura Red de Bibliotecas de LOCAL de Andalucía (ReBJA), y también con algunos bibliotecarios de estas, y de recopilar y examinar diferente documentación que nos ha sido proporcionada, los diseñadores de BD encargados del proyecto hemos sintetizado las informaciones recibidas de la siguiente manera:

1. Servicio de préstamo

- Mediante este servicio, se ofrece la posibilidad de sacar los diversos documentos (libros, revistas, vídeos, CD ...) de la biblioteca.
- Para utilizar el servicio de préstamo, los usuarios deben tener el carné de biblioteca.
- Este carné sirve para utilizar otros servicios de la biblioteca, como el servicio de acceso a Internet, el servicio WiFi, consulta a BD o participación en algunas actividades que organicen las bibliotecas.
- El mismo carné sirve para toda la Red de Bibliotecas y permite disfrutar de ventajas interesantes en el mundo de la cultura.
- La solicitud del carné de biblioteca se debe hacer desde la misma biblioteca o en línea. En el formulario de solicitud, se piden al usuario los datos personales necesarios (nombre y apellidos, dirección, fecha de nacimiento, etc.).
- Además, los usuarios pueden hacer otras gestiones en línea relacionadas con el préstamo: reservas y renovaciones de documentos.

2. Normativa de uso del servicio de préstamo de la Red de Bibliotecas

- Tener el carné implica aceptar las normas de funcionamiento de la biblioteca.
- El carné es personal e intransferible.
- Los usuarios menores de dieciséis años necesitan la autorización de los padres para hacerse el carné, y también los mayores de dieciséis años que no tengan DNI.
- Hay que comunicar, en la biblioteca, cualquier cambio de domicilio o la pérdida del carné.
- Cada lector puede llevarse en préstamo hasta diez documentos entre libros, revistas y audiovisuales, por un plazo de tres semanas, los libros, y una semana, las revistas y audiovisuales.
- Pasado el plazo, cualquier documento en préstamo puede ser renovado, siempre que ningún otro usuario lo haya reservado. La renovación se puede hacer en la biblioteca, por teléfono, por correo electrónico o por Internet.
- Hay que tener cuidado de los documentos prestados. Si un usuario no devuelve los documentos en el plazo fijado, puede ser excluido del servicio de préstamo de las bibliotecas de la Red durante un tiempo equivalente al que se ha retrasado en la devolución. Si un usuario pierde o daña un documento, lo notificará a la biblioteca y tiene que comprar el mismo documento o abonar el importe.
- Quedan excluidos de préstamo los siguientes documentos:
- Algunas obras de referencia: enciclopedias, diccionarios, atlas, etc.
- Fondo de reserva.
- Documentos pertenecientes a la colección local.
- Números corrientes de revistas y de periódicos.
- Los documentos que la biblioteca crea conveniente que no salgan de la biblioteca.
- Además del servicio de préstamo al usuario individual, las bibliotecas ofrecen otros tipos de préstamo:
- Servicio de préstamo interbibliotecario. Mediante el servicio de préstamo interbibliotecario, las bibliotecas de la Red se encargan de localizar y proporcionar los documentos que no tiene el fondo propio y que están disponibles en otras bibliotecas de la Red. El préstamo entre las bibliotecas de la Red tiene un precio público establecido de 1,20 €.
- Servicio de préstamo a domicilio. Este servicio se dirige a personas con problemas de movilidad temporal o permanente, tales como enfermos crónicos, personas en periodo de convalecencia, con discapacidades físicas o personas mayores. No todas las bibliotecas de la Red ofrecen este servicio. El servicio de préstamo a domicilio tiene un precio público establecido de 1,50 €.

3. Consulta del catálogo

- Se debe poder consultar la disponibilidad de los fondos bibliotecarios por los siguientes conceptos:
- Firma (identificador del ejemplar físico objeto de préstamo)

- título
- Palabra clave contenida en el título
- Autor (concretamente, por el apellido)
- Nombre o apellidos incompletos del autor
- materia
- Además, las búsquedas se han de poder limitar a los siguientes ámbitos:
- Una biblioteca concreta
- Todas las bibliotecas de una sola población
- Todas las bibliotecas de una sola comarca
- Finalmente, se deben poder limitar los resultados obtenidos en función de los siguientes conceptos:
- idioma
- Tipo de formato
- Año de publicación

4.4. Identificación de entidades

La especificación de los requerimientos de datos sirve como punto de partida para la elaboración del esquema conceptual de la futura BD. A continuación, lo primero que hay que hacer es identificar las entidades y sus atributos.

La identificación de las entidades y de sus atributos se puede documentar con un listado que siga el siguiente formato:

ENTIDAD1 (Atributo1, Atribut2, ...)

ENTIDAD2 (Atributo1, Atribut2, ...)

....

ENTIDADn (Atributo1, Atribut2, ...)

En el listado, los nombres de las entidades irán con mayúsculas, y los nombres de los atributos comenzarán con mayúsculas.

El ejemplo parte de la información referida en el apartado "Captura y abstracción de los requerimientos de datos".

Siguiendo con el ejemplo BD de la ReBJA, en una primera aproximación, repasando las especificaciones que tenemos, podríamos obtener una solución que contara al menos con las tres entidades siguientes, que incorporan los respectivos atributos expresados entre paréntesis:

- **LOCAL** (Poblacion, Comarca). Posibles atributos adicionales: Nombre, Dirección y Teléfono.
- **DOCUMENTO** (Formato, Importe, ExclosPrestec, Firma, Título, Autor, Materia, Idioma, AñoPublicacion).
- **USUARIO** (DNI, Nombre, Apellidos, Dirección, FechaNacimiento, Carnet, FechaExclusionPrestamos).

Pero estamos en condiciones de depurar este resultado rudimentario. Fijémonos en que habrá muchas poblaciones y comarcas que se repetirán para diferentes LOCAL. Sería mejor, pues, que estos dos atributos constituyeran dos entidades independientes, convenientemente relacionadas con la entidad LOCAL: POBLACION y COMARCA.

Otra ventaja de utilizar estas dos entidades en vez de considerarlas atributos de una entidad concreta, es que las podremos reaprovechar relacionándolas con otras entidades, en caso necesario.

Pasa lo mismo con otros atributos de DOCUMENTO: el formato, el autor la materia y el idioma.

Sería mejor tener una entidad para registrar los tipos de formato, que siempre serán los mismos (libro, revista, CD, etc.), llamada por ejemplo FORMATO, e relacionarse con DOCUMENTO.

Estaremos ante el mismo fenómeno con los idiomas (catalán, castellano, inglés, etc.) y las materias (historia, música, física, etc.). Por lo tanto, será conveniente contar con dos entidades más: IDIOMA y MATERIA.

Por otra parte, los autores podrán ser autores de una pluralidad de documentos y, a fin de simplificar las búsquedas ulteriores por autor, es preferible destinar una entidad propia para representarlos (podemos llamarla, sencillamente, AUTOR).

Después de hacer estas consideraciones, nos encontraríamos con unas cuantas entidades más que inicialmente:

- LOCAL (Nombre, Dirección, Teléfono)
- DOCUMENTO (Firma, Título, AñoPublicacion, Importe, ExcluidoPrestamo)
- USUARIO (Carnet, DNI, Nombre, Apellidos, Dirección, FechaNacimiento, FechaExclusionPrestamos)
- POBLACION (Nombre)
- COMARCA (Nombre)
- FORMATO (Descripcionn)
- AUTOR (Nombre, Apellidos)
- MATERIA (Descripcionn)
- IDIOMA (Descripcionn)

Debemos ser conscientes de que cualquier especificación que no se desprenda directamente de los documentos de trabajo inicial en que hemos sintetizado los diferentes requerimientos de datos detectados como, por ejemplo, añadir los nuevos atributos a LOCAL debe validarse mediante nuevas entrevistas, o el examen de nueva documentación o bien revisión de la antigua, o, si no, observando *in situ* el proceso en ejecución que pueda conllevar una innovación en la especificación.

En cambio, otros aspectos que pueden parecer más osados (como representar lo que era inicialmente el atributo de una entidad como otra entidad independiente) no deben implicar necesariamente unos procesos de validación como los que acabamos de describir, si se tratan puramente de decisiones técnicas de diseño.

Hay que hacer notar que, a poco complicada que sea la porción del mundo real a modelizar, normalmente es posible obtener varios modelos conceptuales, en algunos casos alternativos y equivalentes, y en otros orientados a satisfacer en mayor o menor medida diferentes finalidades, pero con resultados igualmente correctos.

4.5. Designación de relaciones

Tras la captura y abstracción de los requerimientos de datos y de la identificación de entidades, el siguiente paso consiste en establecer las relaciones necesarias entre las entidades detectadas.

Las relaciones se pueden documentar con el siguiente formato:

Relación (Atributo 1, Atributo2, ...), entre ETIDAD1 y ETIDADn

En que el nombre de la Relación comenzará con mayúsculas y, en caso de disponer de atributos, estos irán entre paréntesis y con la inicial del nombre también con mayúsculas. Se deberá especificar el nombre de las entidades que relaciona.

El ejemplo mencionado se refiere a las informaciones contenidas en el apartado "Captura y abstracción de los requerimientos de datos", y se desarrolla a lo largo de las diferentes fases de diseño conceptual de BD.

En algún caso, las relaciones pueden incorporar algún atributo (como ocurre con la Relación **Préstamo**), al igual que las entidades. Inicialmente podemos encontrar, al menos, cuatro relaciones que sólo afectan a las tres entidades originarias:

- **Préstamo** (Fecha, Tipo, Precio), entre USUARIO y DOCUMENTO
- **RenovacionPrestamo** (Fecha), entre USUARIO y DOCUMENTO
- **Reserva** (Fecha), entre USUARIO y DOCUMENTO
- **Ubicación**, entre LOCAL y DOCUMENTO

Pero, después de considerar el establecimiento del resto de entidades, tendremos que añadir las relaciones correspondientes:

- **Esta**, entre LOCAL y POBLACION
- **Pertenece**, entre POBLACION y COMARCA
- **TE1**, entre FORMATO y DOCUMENTO
- **TE2**, entre AUTOR y DOCUMENTO
- **TrataDe**, entre MATERIA y DOCUMENTO
- **Expresado**, entre IDIOMA y DOCUMENTO

Notamos que cuando se repite un mismo nombre en más de una Relación (como aquí ocurre con Te), hay que numerarlas correlativamente, para evitar confusiones al hacer referencia a cada una.

Y, además, podríamos considerar el establecimiento de una Relación entre USUARIO y POBLACION para completar su dirección debidamente (como hemos hecho con los LOCAL). Quedaría así:

- **Vive**, entre USUARIO y POBLACION

4.6. Establecimiento de claves

Recordemos que la **clave primaria** de una entidad está constituida por un atributo, o por un conjunto de atributos, los valores de los que son capaces de identificar unívocamente las instancias de aquella.

A veces, una entidad dispone de más de un atributo, o conjunto de atributos, que están en condiciones de constituir la clave primaria. En este caso, hablamos de **claves candidatas**.

Y una vez que el diseñador elige un atributo o conjunto de atributos concretos para identificar unívocamente las instancias, entre los que cumplen las condiciones (**claves candidatas**), este o estos atributos pasan a ser la **clave primaria**, y los no seleccionados permanecen como **claves alternativas**.

Un criterio importante a la hora de decidirse porque uno o más atributos formen la clave primaria de una entidad, es que su valor no cambie nunca o, cuando menos, muy raramente.

El atributo o conjunto de atributos que forman la **clave primaria** deben aparecer **subrayados** en la documentación.

Por ejemplo, no sería muy prudente decidirse por el número de teléfono móvil como clave primaria de una entidad que almacena personas, porque, a pesar de ser un número habitualmente de uso personal, puede cambiar a lo largo del tiempo (por ejemplo, una persona puede regalar su móvil a otra). En cambio, el número de DNI puede ser, en general, una buena opción, ya que, en principio, este número es personal e intransferible.

El ejemplo mencionado se refiere a las informaciones contenidas en el apartado "Captura y abstracción de los requerimientos de datos", y se desarrolla a lo largo de las diferentes fases de diseño conceptual de BD.

Continuando con nuestro ejemplo, estamos en condiciones de encontrar las claves, formadas por sólo un atributo, las cuales se muestran subrayadas:

- DOCUMENTO (Signatura, Título, AñoPublicacion, Importe, ExcluidoPrestamo)
- COMARCA (Nombre)
- POBLACION (Nombre)

- FORMATO (Descripción)
- MATERIA (Descripción)
- IDIOMA (Descripción)

La entidad AUTOR no tiene ningún atributo que identifique plenamente sus instancias. Aquí podríamos haber optado por combinar los valores que adopten los atributos Nombre y Apellidos en cada tupla, para no tener que introducir ningún tipo de codificación artificial, pero ésta no deja de ser una opción arriesgada, ya que puede ocurrir que haya dos autores con el mismo nombre y apellidos. Por lo tanto, añadimos un nuevo atributo, llamado Código, para que no haya problemas con la clave primaria de esta entidad. Por tanto, la entidad nos queda así:

- AUTOR (Código, Nombre, Apellidos)

En cierto modo podríamos considerar que USUARIO tiene dos claves alternativas: Carnet y DNI. Pero en realidad no es así, ya que puede haber usuarios menores de edad que aún no tienen DNI. Y ya sabemos que ninguna clave primaria puede admitir valores nulos, ya que entonces no serviría para distinguir unívocamente las instancias entre ellas. En cambio, todo usuario dispondrá de un número de carné. Por lo tanto, nos quedará la estructura siguiente:

- USUARIO (Carnet, DNI, Nombre, Apellidos, Dirección, FechaNacimiento, FechaExclusionPrestamos)

Finalmente, debemos examinar la entidad LOCAL. El nombre de los institutos se puede repetir, y de hecho se repite (por ejemplo, hay una LOCAL llamado Baños y Goles en Sevilla, otro en Coria del Río, otro en Espartinas, etc.). Por lo tanto, el atributo Nombre no es suficiente para constituir por sí solo la clave primaria de la entidad. Una opción sería establecer algún tipo de codificación. Pero, en este caso, hemos preferido considerar LOCAL como una entidad débil que depende de POBLACION. Por lo tanto, la clave primaria de LOCAL estará formada por la clave primaria de la entidad fuerte (el atributo Nombre de POBLACION) más el atributo discriminante de la entidad débil (el atributo Nombre de LOCAL), que también mostramos subrayado:

- LOCAL (Nombre, Dirección, Teléfono)

4.7. Establecimiento de cardinalidades

El establecimiento correcto de las cardinalidades adecuadas para cada Relación depende de las características del mundo real que se quieren modelizar.

Hay que añadir, pues, la información referente a las cardinalidades de las relaciones en la documentación de diseño.

El ejemplo se plantea en el apartado "Captura y abstracción de los requerimientos de datos", y se va construyendo siguiendo las diferentes fases de diseño conceptual de BD.

En el modelo que estamos construyendo (siguiendo el ejemplo de la BD de la ReBJA) podemos establecer las cardinalidades siguientes:

- ✓ Un mismo usuario puede tomar prestados diferentes documentos, y un mismo documento se puede prestar a diferentes usuarios, a lo largo del tiempo:

Préstamo (Fecha, Tipo, Precio), entre USUARIO y DOCUMENTO: M-N

- ✓ Pasa lo mismo en caso de renovar un préstamo o de reservar un documento:

PidePrestado (Fecha), entre USUARIO y DOCUMENTO: M-N

Reserva (Fecha), entre USUARIO y DOCUMENTO: M-N

- ✓ Todo documento pertenecerá a la biblioteca de un LOCAL concreto, la que podrá disponer de muchos documentos:

Ubicación, entre LOCAL y DOCUMENTO: 1-N

- ✓ Dentro de una misma población podrán coexistir diferentes LOCAL, pero un LOCAL estará en una población concreta. Además no olvidemos que, al tratarse de una entidad débil, la cardinalidad debe ser 1-N, con la entidad débil junto a la N:

Esta, entre LOCAL y POBLACION: N-1

- ✓ Cada población pertenece a una sola comarca, pero cada comarca tendrá más de una población dentro de su territorio:

Pertenece, entre POBLACION y COMARCA: N-1

- ✓ Se ha considerado que cada documento sólo tiene un formato, pero evidentemente cada formato será aplicable a muchos documentos diferentes:

TE1, entre FORMATO y DOCUMENTO: 1-N

- ✓ Con una cardinalidad M-N, hacemos posible registrar más de un autor para un mismo documento:

TE2, entre AUTOR y DOCUMENTO: M-N

- ✓ En cambio, consideramos que cada documento sólo puede Tratar sobre una sola materia:

TrataDe, entre MATERIA y DOCUMENTO: 1-N

- ✓ Un documento (como una película en DVD) podrá estar *expresado* en diferentes idiomas:

Expresado, entre IDIOMA y DOCUMENTO: M-N

- ✓ Cada usuario tiene su domicilio en una población concreta. Pero en una misma población pueden vivir diferentes usuarios:

Vive, entre USUARIO y POBLACION: N-1

4.8. Restricciones de participación y límites de cardinalidad

Se dice que la **participación de una entidad en una Relación** es **total** si cada una de sus instancias participa una vez, al menos, en la Relación mencionada, y **parcial** de lo contrario.

- Préstamo (Fecha, Tipo, Precio), entre USUARIO (parcial) y DOCUMENTO (parcial): M-N
- PidePrestado (Fecha), entre USUARIO (parcial) y DOCUMENTO (parcial): M-N
- Reserva (Fecha), entre USUARIO (parcial) y DOCUMENTO (parcial): M-N
- Ubicación, entre LOCAL (parcial) y DOCUMENTO (total): 1-N
- Esta, entre LOCAL (total) y POBLACION (parcial): N-1
- Pertenece, entre POBLACION (total) y COMARCA (total): N-1
- TE1, entre FORMATO (parcial) y DOCUMENTO (total): 1-N
- TE2, entre AUTOR (total) y DOCUMENTO (total): M-N
- TrataDe, entre MATERIA (parcial) y DOCUMENTO (total): 1-N
- Expresado, entre IDIOMA (total) y DOCUMENTO (total): M-N
- Vive, entre USUARIO (total) y POBLACION (parcial): N-1

Además, hemos de establecer un límite máximo a la cardinalidad de la Relación Préstamo para que, en principio, un usuario no pueda solicitar en préstamo más de diez documentos simultáneamente.

Las relaciones quedarán documentadas, pues, con el siguiente formato:

Relación (Atributo1, Atributo2, ...), entre ENTIDADi (participación) y ENTIDADj (participación): tipo_de_cardinalidad

En que el nombre de la Relación comenzará con mayúsculas y, en caso de disponer de atributos, estos irán entre paréntesis, separados por comas, y con la inicial del nombre también con mayúsculas. Se deberá especificar el nombre de las entidades que relaciona y el tipo de cardinalidad, que puede ser 1-1, 1-N, N-1 o M-N. La participación puede ser total o parcial.

4.9. Elaboración de un esquema conceptual

Una vez tenemos los requerimientos; y hemos identificado las entidades, atributos e relaciones; y gracias al conocimiento logrado de las necesidades que deben satisfacer los datos, el diseñador está en condiciones de elaborar un esquema conceptual completo de estas y de sus relaciones.

Para confeccionar este modelo conceptual, el diseñador de BD utilizará algún modelo de datos que se adapte a las necesidades del proyecto, y permita continuar trabajando en él durante la fase ulterior de diseño lógico.

Un diagrama ER es un esquema gráfico que sirve para representar un modelo ER.

El modelo de datos más utilizado para elaborar el esquema conceptual es el modelo ER, y la documentación del diseño conceptual culmina con el **diagrama ER**.

UML

Acrónimo de unified modeling language (lenguaje unificado de modelización). Notación gráfica que permite especificar datos y aplicaciones orientadas a objetos. Se entiende por modelo de datos UML aquella parte del UML destinada a describir los datos.

Aunque el modelo ER sigue siendo el modelo de datos más utilizado, con diferentes variaciones en aspectos de notación, cada vez se utiliza más el modelo UML de datos, muy útil en proyectos que basen la implementación de la programación en el paradigma de la orientación a objetos.

Para empezar a trabajar con los diagramas ER, puede ser cómodo establecer el esqueleto del modelo, sin incluir aún todos los detalles (atributos, cardinalidades, etc.). En la [figura 1](#), hay un esquema conceptual inacabado, expresado con la ayuda del modelo de datos ER, que sólo incluye las entidades y las relaciones detectadas inicialmente.

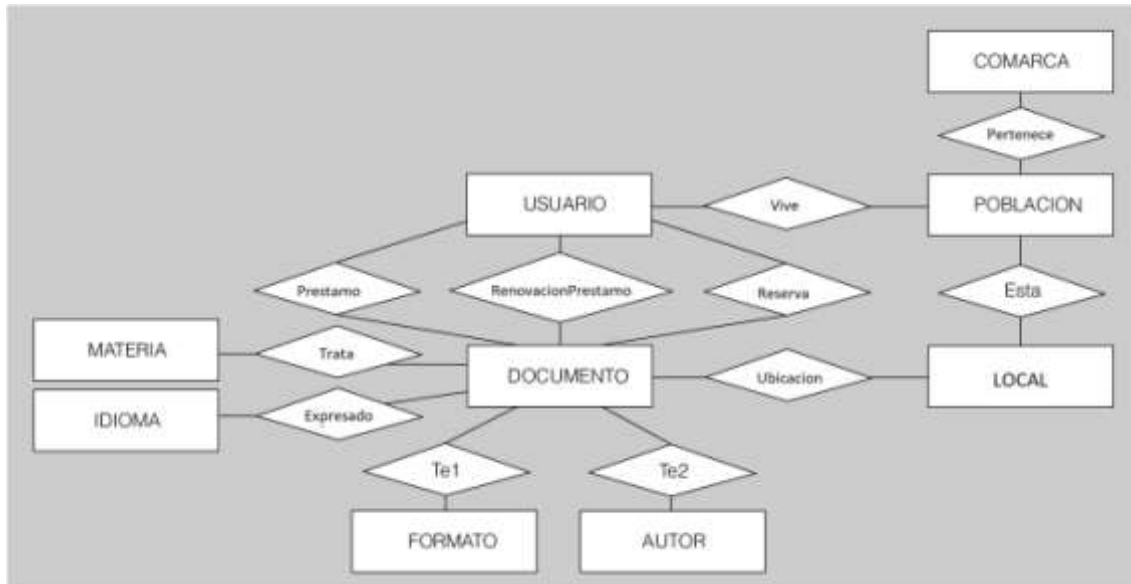


Figura 19 Diagrama ER inicial de la ReBJA

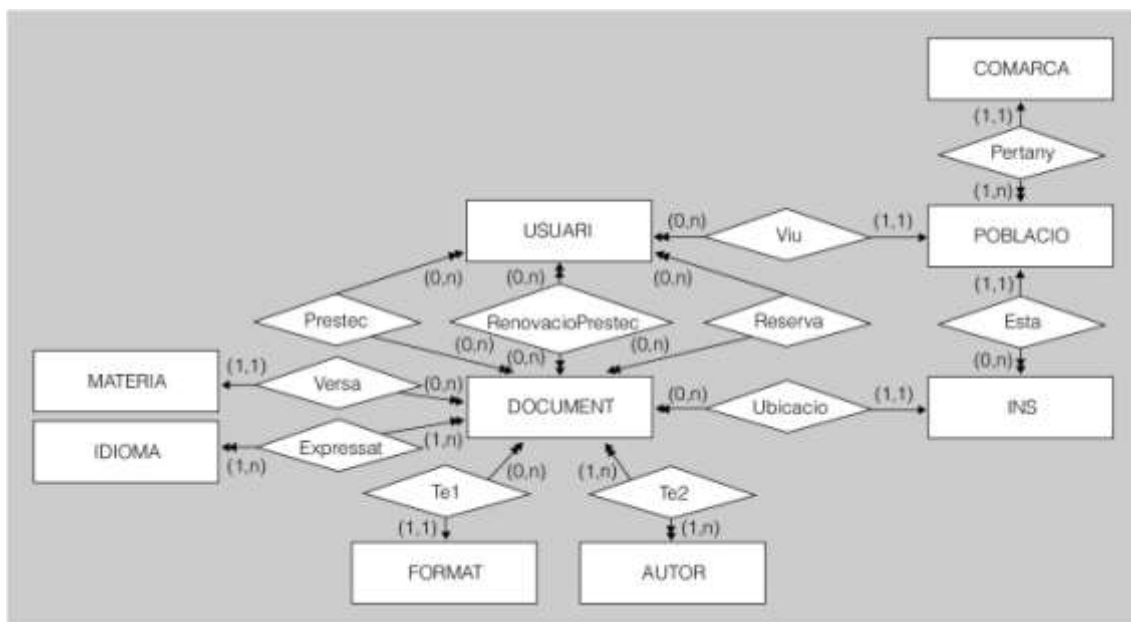


Figura 20 Diagrama ER de la ReBJA a medias

A la hora de diseñar el diagrama ER, puede ser una buena práctica partir de un primer diagrama inacabado (como el de la figura 1) e incorporar más detalles progresivamente. En la figura 2, por ejemplo, ya han quedado establecidas las cardinalidades y las restricciones de participación.

Y el último paso que hará el diseñador para completar el diagrama, de manera que refleje todos los requerimientos previamente detectados, se puede ver en el ejemplo de la figura 3, en la que hay especificados todos los atributos y claves.

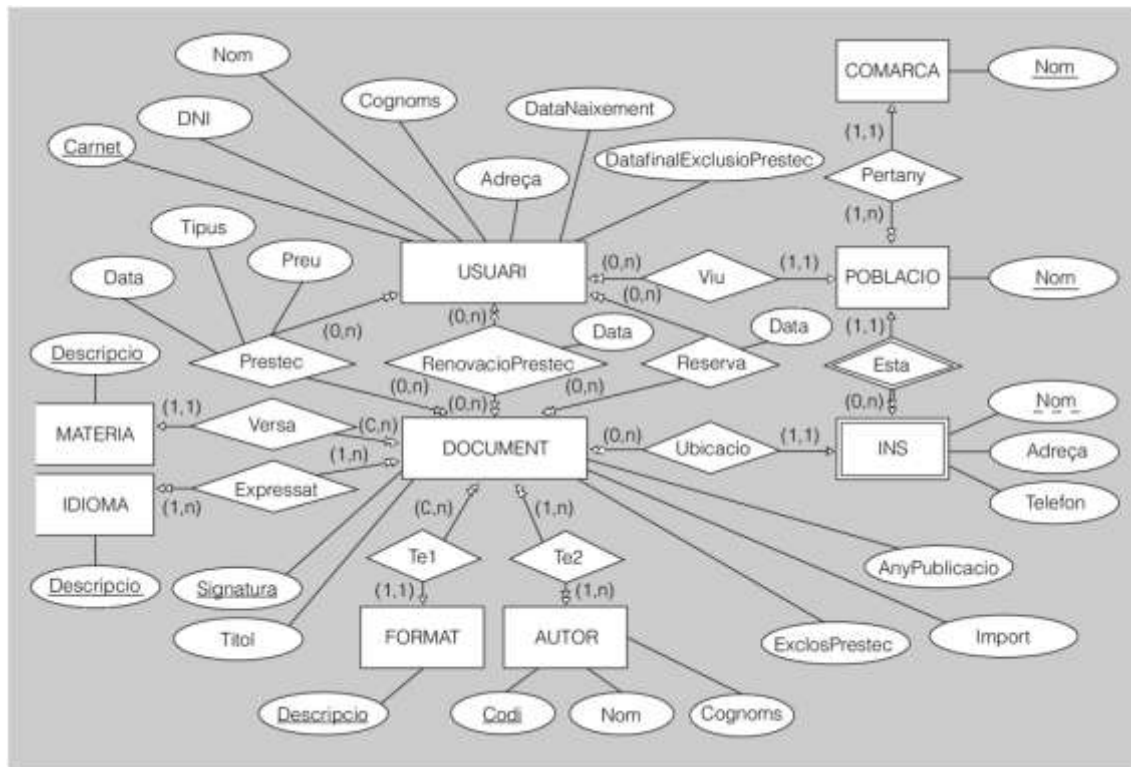


Figura 21 Diagrama ER de la ReBJA totalmente terminado

5. EXTENSIONES DEL MODELO ENTIDAD-RELACIÓN

Las estructuras básicas del modelo Entidad-Relación (modelo ER) permiten representar la mayoría de situaciones del mundo real que habitualmente hay que incorporar en las BD. Pero, a veces, ciertos aspectos de los datos se describirán mediante unas construcciones más avanzadas del modelo ER, las cuales comportan una extensión del modelo ER básico. Estas ampliaciones del modelo ER consisten en la especialización, la generalización y la agregación, de entidades.

5.1. Especialización y generalización

Nos podemos encontrar con el caso de alguna entidad tipo en que, además de las características generales, comunes a todas sus instancias nos interese modelizar, adicionalmente, ciertas características específicas aplicables sólo a parte de sus instancias.

Entonces, podremos considerar que esta entidad tipo contiene otras entidades tipo, de nivel inferior, con características propias.

La **especialización** permite reflejar la existencia de una entidad general, denominada **entidad superclase**, que se puede especializar en diferentes entidades subclase.

La **entidad superclase** permite representar las características comunes de la entidad desde un punto de vista general. Las **entidades subclase**, en cambio, permiten representar las características propias de las especializaciones de la entidad superclase.

Las instancias de las subclases deben ser, al mismo tiempo, instancias de la superclase respectiva.

El proceso de designación de subclases a partir de una superclase llama **especialización**.

Ejemplo de especialización

Hasta ahora contábamos con una única entidad PROFESOR, que nos servía para trabajar con todos los docentes del centro, ya que aún no habíamos detectado ningún subconjunto de este colectivo que nos hubiera hecho pensar en implementar una especialización.

Pero resulta que la dirección del centro quiere implicar, en la gestión de éste y en su mantenimiento informático, el profesorado de dos familias profesionales: la administrativa y la informática, respectivamente.

Por lo tanto, nos interesa tener constancia, por un lado, de la titulación de los profesores de la familia administrativa y de su especialidad, ya que en función de estas características podrán asumir, o no, las responsabilidades que se les quieren encomendar.

También puede resultar útil saber cuál es la especialidad principal, tanto en hardware como en software, del profesorado de la familia de informática, para asignar las tareas de mantenimiento con una cierta garantía de éxito.

Como consecuencia de todo ello, implementaremos una especialización de la entidad PROFESOR en dos subclases: ADMINISTRATIVO, que incorporará dos nuevos atributos (titulación y especialidad), y INFORMatico, que incorporará otros dos (Sistemas y Programación).

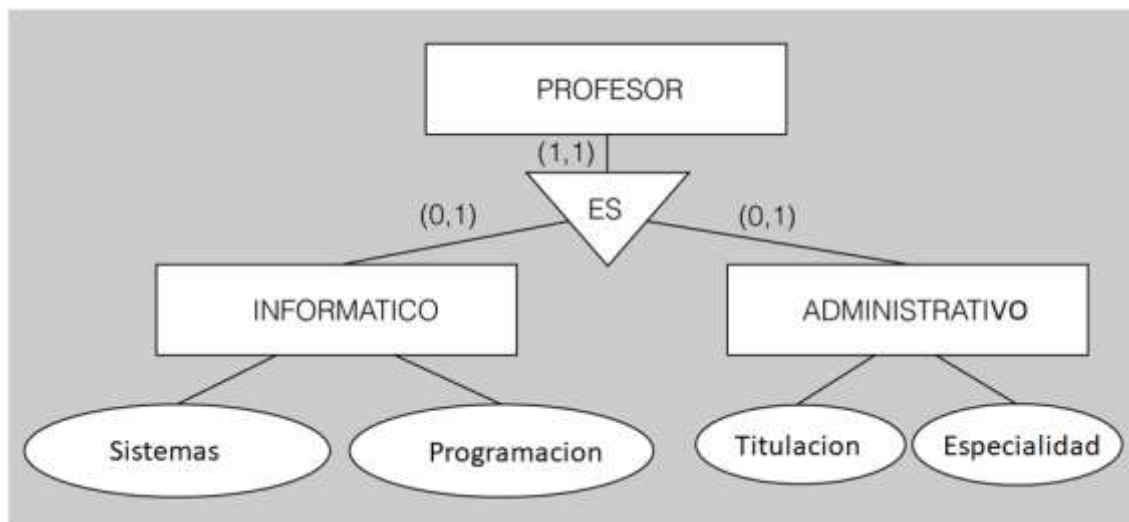


Figura 22 Ejemplo de especialización

La especialización en los diagramas ER se representa con un triángulo.

La especialización, pues, permite reflejar las diferencias entre las instancias de una misma entidad, mediante el establecimiento de diferentes entidades de nivel inferior, las cuales agrupan a los subconjuntos de instancias con características específicas comunes.

Estas características propias de las subclases pueden consistir tanto en la existencia de atributos como en la participación en relaciones, pero en ningún caso pueden ser de aplicación a todas las instancias de la superclase considerada como tal.

La **generalización**, en cambio, es el resultado de observar cómo diferentes entidades preexistentes comparten ciertas características comunes (es decir, identidad de atributos o de relaciones en las que participan).

En función de las similitudes detectadas entre diferentes entidades, estas se pueden llegar a sintetizar en una sola entidad, de nivel superior, mediante un proceso de generalización.

La generalización sirve para resaltar las similitudes entre entidades, por encima de las diferencias, y también para simplificar las representaciones de los datos, al evitar la repetición de atributos compartidos por diferentes subclases.

Ejemplo de generalización

Hasta ahora, hemos utilizado dos entidades diferentes que nos han servido para modelar dos categorías, también diferentes, existentes en el mundo real: ALUMNO y PROFESOR.

Pero, es evidente que tanto los alumnos como los profesores son personas, aunque con roles diferentes. Por lo tanto, tendrán una serie de características comunes, que se podrán modelizar de la misma manera.

Así, tanto los unos como otros tendrán nombre, apellidos, teléfonos de contacto, etc., que se podrán modelizar mediante los mismos atributos.

También es posible que las dos tipologías puedan participar en las mismas relaciones. Por ejemplo, para indicar la localidad de residencia, lo más habitual es relacionar la entidad que representa a las personas con otra entidad que almacena las diferentes localidades.

En definitiva, partiendo de las entidades ALUMNO y PROFESOR, podríamos crear otra entidad, superclase de las anteriores, y llamarla, por ejemplo, PERSONA.

De esta manera implementar la entidad PERSONA, como generalización de ALUMNO y PROFESOR, que contendrá los atributos comunes a sus subclases, y además participará directamente en las relaciones que también sean comunes a las subclases mencionadas.

*La **generalización** en los diagramas ER se representa con un **triángulo**, como en la especialización.*

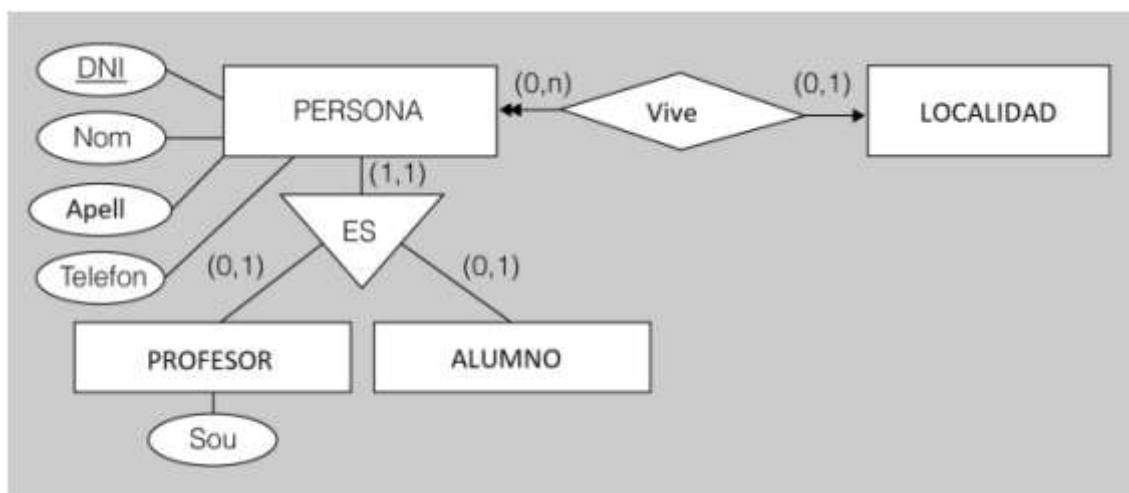


Figura 23 Ejemplo de generalización

El producto resultante de la especialización y de la generalización es, pues, idéntico. La diferencia entre ambos reside en el tipo de proceso que conduce a cada una:

- La especialización deriva de un proceso de diseño descendente, durante el cual, a partir de una entidad preexistente, considerada como superclase se detecta la utilidad de establecer ciertas subclases, debido a la existencia de ciertas características (atributos y participaciones en relaciones) no aplicables a todas las instancias de la superclase.
- La generalización responde a un proceso considerado de diseño ascendente. Durante este tipo de diseño se valora la utilidad de contemplar varias entidades preexistentes, llamadas subclases, dependientes de una misma superclase común a todas ellas. La superclase presenta unas características comunes (atributos y participaciones en relaciones) a todas las subclases que dependen.

Herencia de propiedades

Tanto en el caso de generalización como en el de especialización, las características de la entidad superclase extienden hacia las entidades subclase. Como ya sabemos, estas características pueden consistir bien en atributos de la entidad superclase, o bien en su participación en diferentes relaciones.

Llamamos **herencia de propiedades** la transmisión de características (atributos e relaciones) desde la entidad superclase hacia las entidades subclase.

Puede suceder que una misma entidad adopte el rol de subclase en un proceso de generalización o especialización y que, al mismo tiempo, asuma el papel de superclase en otro de estos procesos en que participe.

Jerarquía de entidades

Cuando se encadenan diferentes generalizaciones o especializaciones de tal manera que una misma entidad es subclase de una estructura, y superclase de otra, tiene lugar lo que se llama jerarquía de entidades.

Cuando se produce una jerarquía de entidades, las entidades de los niveles inferiores pueden heredar características no sólo de la superclase respectiva, sino también de otras clases de niveles superiores.

Llamamos **herencia múltiple** a la recepción, por parte de una entidad subclase, tanto de las características (atributos e relaciones) de su superclase, como de las de otras entidades de niveles superiores, dentro de una estructura jerárquica de entidades con generalizaciones o especializaciones encadenadas.

Ejemplo de jerarquía de entidades y de herencia múltiple

Como resultado de un proceso de generalización hemos conferido, a la entidad PERSONA, la calidad de superclase de las entidades PROFESOR y ALUMNO, consideradas subclases de aquella.

Al mismo tiempo, pero como resultado de un proceso de especialización, hemos conferido a la entidad PROFESOR la categoría de superclase de dos nuevas entidades, subclases de la misma: INFORMATICO y ADMINISTRATIVO.

A consecuencia de todo ello, las entidades del nivel inferior (INFORMATICO y ADMINISTRATIVO) no sólo heredarán las características de su superclase (PROFESOR), sino también las de las otras entidades de niveles superiores de las que sean descendientes y, por tanto, herederas.

En este caso, pues, INFORMATICO y ADMINISTRATIVO heredarán las características de su superclase (PROFESOR) y también las propiedades de la superclase de aquella

(PERSONA). Pero no heredarán ninguna propiedad de ALUMNO, porque no son descendientes de esta entidad.

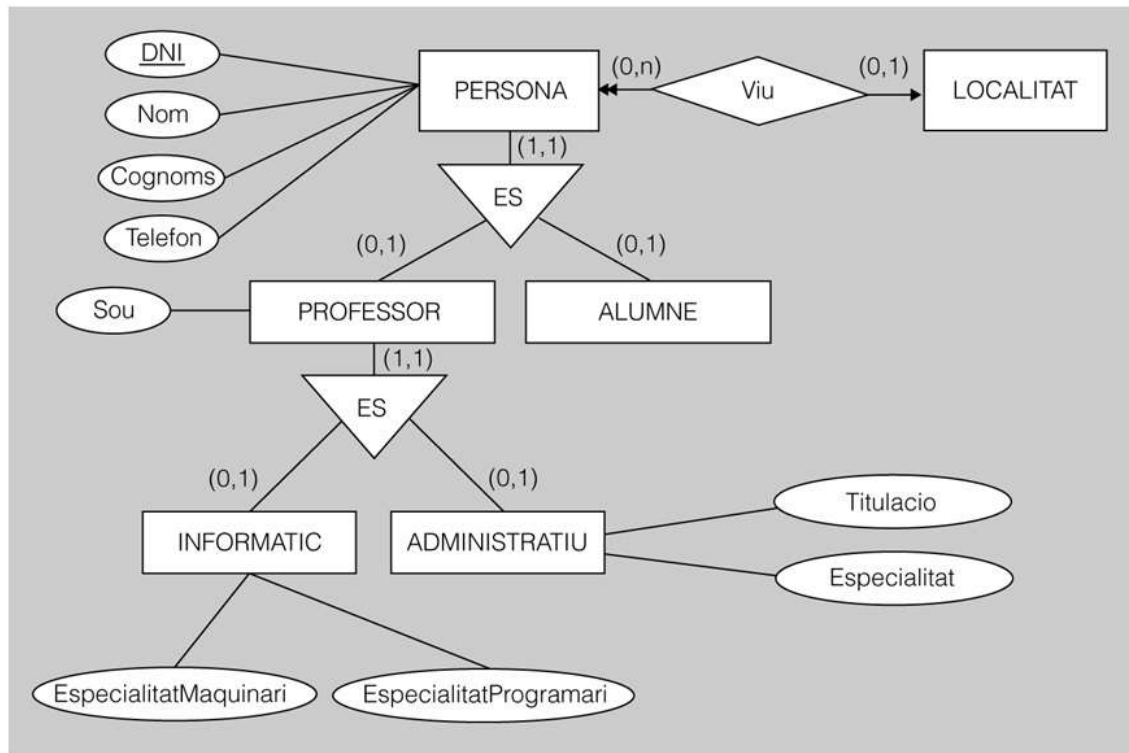


Figura 24 Ejemplo de herencia múltiple

Restricciones

Con el fin de modelizar más exactamente la parcela del mundo real que nos interese, se pueden establecer ciertas restricciones sobre las especializaciones o generalizaciones detectadas.

Un primer tipo de restricciones define si las instancias pueden pertenecer simultáneamente o no a más de una subclase de una estructura simple (es decir, que cuente con una sola superclase y un solo nivel de subclases) de generalización o especialización. En estos casos, las entidades de tipo subclase pueden ser de dos tipos:

- **Disjuntas** . Una misma entidad instancia no puede aparecer en dos entidades subclase diferentes. Se representa en el diagrama añadiendo una etiqueta con la letra D.
- **Solapadas** . Una misma entidad instancia puede aparecer en dos (o, incluso, en más de dos) entidades subclase diferentes. Se representa en el diagrama añadiendo una etiqueta con la letra S.

Un segundo tipo de restricciones especifica si toda instancia de la superclase debe pertenecer simultáneamente a una o más de las subclases o no. Aquí las entidades de tipo subclase también pueden ser de dos tipos:

- **Totales** . Toda instancia de la entidad superclase debe pertenecer simultáneamente, como mínimo, a una de sus entidades subclase. Se denota con la etiqueta T.
- **Parciales** . Algunas instancias de la entidad superclase pueden no pertenecer simultáneamente a ninguna de sus entidades subclase. Se denota con la etiqueta P.

Combinando estas restricciones obtenemos, pues, cuatro posibilidades aplicables a las subclases de una generalización o especificación. Hay que separar las letras que se incluyen en la etiqueta con una coma:

- D, T (disjuntas y totales)
- D, P (disjuntas y parciales)
- S, T (solape y totales)

- S, P (solape y parciales)

Ejemplo de subclases D, T

Tendremos que considerar disjuntas las subclases de PERSONA si los reglamentos de funcionamiento del centro no permiten que ningún profesor se matricule como alumno, simultáneamente con el ejercicio de su labor docente.

Al mismo tiempo, las consideraremos totales si nuestra BD registra exclusivamente los datos de profesores y de alumnos, sin ocuparse de otras categorías de personas (como podría ser el personal administrativo, de mantenimiento, de limpieza, etc.).

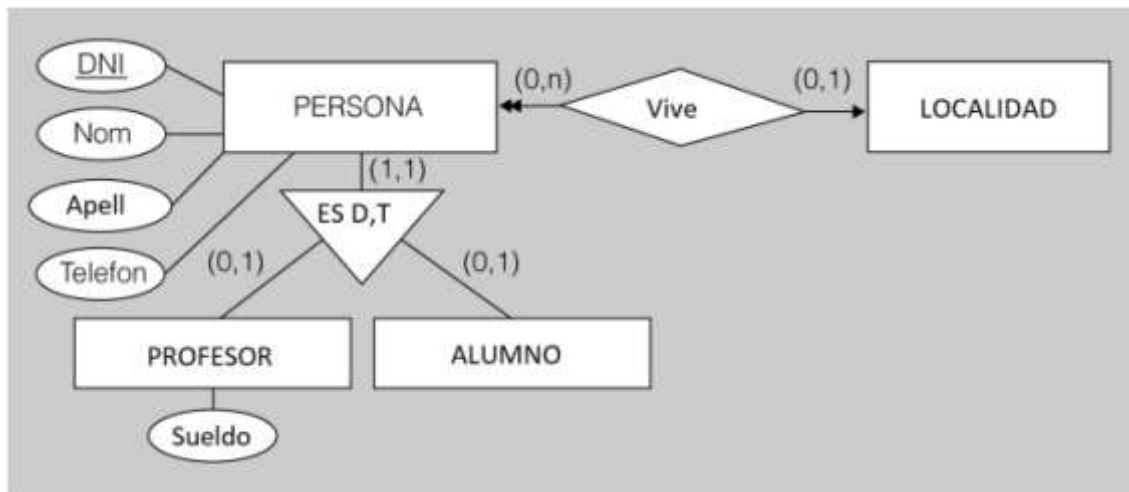


Figura 25 Ejemplo de subclases D, T

Ejemplo de subclases S, P

Tendremos que considerar solapadas las subclases de PROFESOR si queremos reflejar el hecho de que algunos profesores, ejerciendo como tales con una especialidad concreta en un curso académico, pueden tener otras especialidades. Por lo tanto, un profesor podrá ser simultáneamente INFORMATICO y ADMINISTRATIVO.

Por otra parte, las consideraremos parciales porque en nuestro instituto podrá haber, con toda seguridad, profesores de otras especialidades (como electrónicas, comerciales, etc.), que no serán ni informáticos ni administrativos.

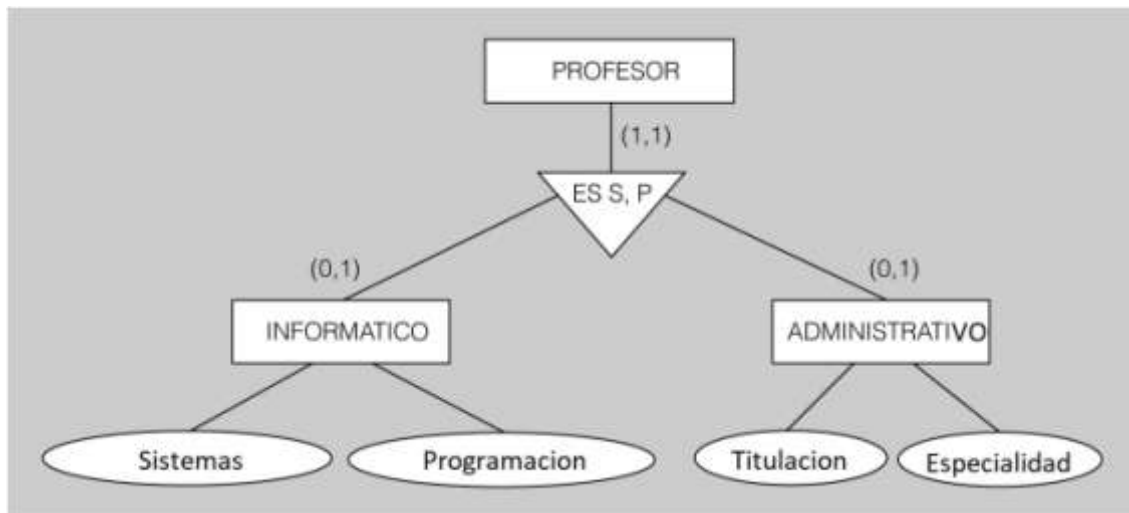


Figura 26 Ejemplo de subclases E, P

Para reflejar una combinación de características aún más complicada, se debe recurrir a una especificación textual que acompañe el diagrama.

Ejemplo de subclases con diferentes restricciones

Imaginemos que queremos añadir una nueva subclase de PERSONA, con el fin de incluir el personal de administración y servicios del centro. Llamaremos esta nueva entidad ADMINISTRATIVO.

Ya hemos expuesto más arriba que los reglamentos internos de nuestro instituto no permiten que ningún profesor sea, simultáneamente, alumno del centro. Pero ahora nos encontramos con que, al personal de administración y servicios, sí se le permite matricularse como alumno en alguno de los estudios impartidos en el centro al tiempo que ejercen su labor profesional.

Las tres subclases serán totales, como antes, porque todo el mundo deberá pertenecer a alguna de las tres categorías reflejadas.

PROFESOR y ALUMNO serán disjuntas entre ellas, al igual que PROFESOR y ADMINISTRATIVO, ya que no se pueden compatibilizar las condiciones mencionadas. Pero, al mismo tiempo, ALUMNO y ADMINISTRATIVO serán solapadas, para que una persona podrá estar incluida en estas dos categorías simultáneamente, según hemos visto.

Para reflejar una realidad como ésta, no hay más remedio que utilizar una especificación textual que, acompañando el diagrama, aclare debidamente las características específicas de cada subclase o agrupación de estas.

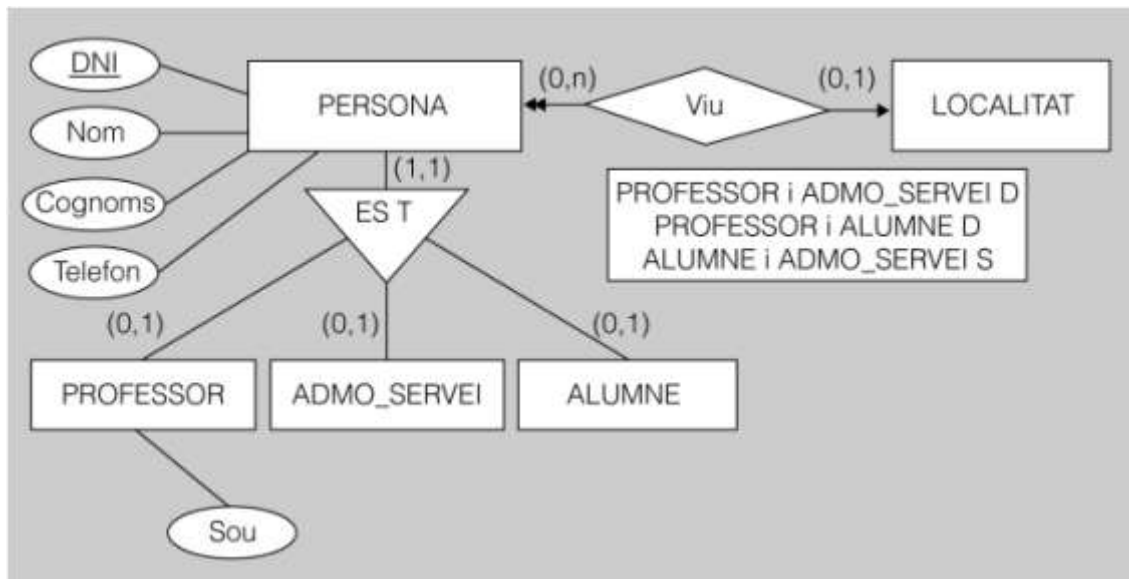


Figura 27 Ejemplo de subclases con diferentes restricciones

Notación

Tanto la especialización como la generalización se representan mediante un triángulo que incluye, en su interior, la etiqueta SE. Esta etiqueta indica que toda instancia de cualquiera de las subclases es, al mismo tiempo, una instancia de la superclase correspondiente (por ejemplo, tanto un informático como un administrativo serán, al mismo tiempo, un profesor).

Para distinguir claramente la superclase en los casos en que hay un gran número de entidades subclase implicadas en la estructura, o bien cuando resulta difícil, por las características del diagrama, alinear claramente todas las subclases, es conveniente indicar los límites de cardinalidad de la generalización o especialización. Para ello, basta con añadir una etiqueta del tipo **mín..màx**, para expresar los límites respectivos, junto a la línea que une cada entidad con el triángulo que representa la generalización o especialización, en el que mín y màx podrán tener los siguientes valores:

- 1..1 en la línea que enlaza la superclase, para que toda instancia de cualquier subclase siempre constituirá, simultáneamente, una y sólo una instancia de la superclase.
- 0..1 en la línea que enlaza cada subclase, porque no necesariamente toda instancia de la superclase deberá ser, simultáneamente, instancia de la subclase en cuestión (lo podrá ser de otra subclase, o podrá no serlo de ninguna, si estamos en presencia de una restricción de parcialidad).

Las entidades que forman parte de una estructura de generalización o especialización se representan como el resto de entidades: cada una con un rectángulo que incorpora el nombre respectivo, y los atributos respectivos cercados dentro de elipses ligadas a su entidad con una línea. Si los atributos forman una clave primaria, su nombre deberá ir subrayado.

En términos de notación diagramática, no establece ninguna diferencia entre una generalización y una especialización. Las diferencias entre ambos fenómenos se reducen al proceso que se ha seguido para derivar en cada uno de ellos, pero no en el resultado, que siempre es el mismo: el establecimiento de una superclase y de unas subclases con unas restricciones concretas, que se representan añadiendo, en la etiqueta sE, las iniciales de las dos restricciones aplicables separadas por una coma:

- D, T (disjuntas y totales)
- D, P (disjuntas y parciales)
- S, T (solape y totales)
- S, P (solape y parciales)

5.2. Agregaciones de entidades

Con las reglas básicas del modelo ER, sólo se pueden modelizar relaciones en que participan exclusivamente entidades, pero no se puede expresar la posibilidad de que una Relación participe directamente en otra Relación. Pero hay un mecanismo, llamado *agregación*, que permite superar la limitación descrita anteriormente, considerando una Relación entre entidades como si fuera una entidad, y utilizándola como tal.

Las agregaciones también son conocidas como entidades asociativas.

La **agregación de entidades** es una abstracción, mediante la cual, una Relación se trata como una entidad de nivel más alto, que agrupa las entidades relacionadas gracias a ella. La agregación debe tener el mismo nombre que la Relación sobre la que se define.

La utilidad de una agregación de entidades, pues, consiste en que la Relación en que se basa se puede relacionar con otras entidades. Una agregación de entidades se denota recuadros todas las entidades que participan en una Relación determinada, a fin de construir una nueva entidad que puede establecer las propias relaciones.

Ejemplo de agregación de entidades

Consideramos la Relación Practica, binaria y de cardinalidad NM, que tiene lugar entre las entidades ALUMNO y DEPORTE, que ya conocemos.

Imaginemos ahora que se quiere tener constancia del profesor que, en su caso, se dedica a entrenar un alumno que practica un deporte determinado. Y recordemos que ya existe en nuestro modelo una entidad llamada PROFESOR.

Una alternativa para representar esta realidad consistiría en crear una Relación ternaria, llamada, por ejemplo, Entrena, entre las entidades ALUMNO, DEPORTE y PROFESOR (figura 10).

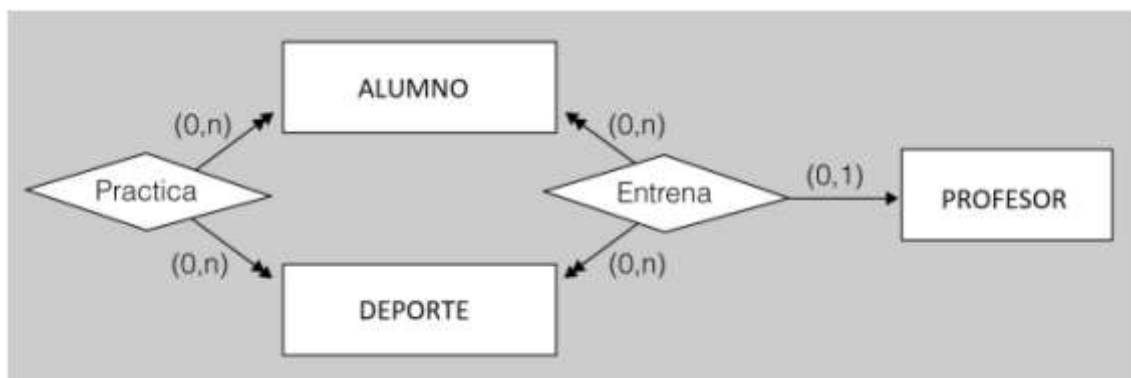


Figura 28 Ejemplo de relaciones redundantes

De este modo, puede parecer que las relaciones Practica y Entrena se pueden combinar en una única Relación. Pero esto no es del todo cierto, ya que habrá relaciones entre ALUMNO y DEPORTE que no dispondrán necesariamente de un profesor que actúe como entrenador.

Ahora bien, hay información redundante en el esquema propuesto hasta ahora, ya que toda combinación entre instancias de las entidades ALUMNO y DEPORTE que hay en Entrena también es Practica.

Si el entrenador sólo fuera un valor, nos podríamos plantear simplemente añadir un atributo a la Relación Practica, que se dijera, por ejemplo, Entrenador. Pero al existir una entidad (PROFESOR) que contiene la instancia aplicable a cada caso, cuando es necesario, debemos descartar esta posibilidad.

Así pues, la mejor manera de reflejar todas estas circunstancias es hacer uso de una agregación de entidades. En este caso, hay que considerar la Relación Practica, entre ALUMNO y DEPORTE, como otra entidad de nivel más alto, llamada PRACTICA. Y, seguidamente, se puede establecer una Relación binaria con cardinalidad 1N entre PROFESOR y la agregación PRACTICA, y llamarla Entrena, y que incluya las combinaciones necesarias entre ambas, para modelizar quien entrena la práctica de los deportes por parte los alumnos, cuando se produce esta circunstancia (figura 11).

Las agregaciones de entidades en los diagramas ER se representan como una agrupación rectangular de las entidades y relaciones que integran.

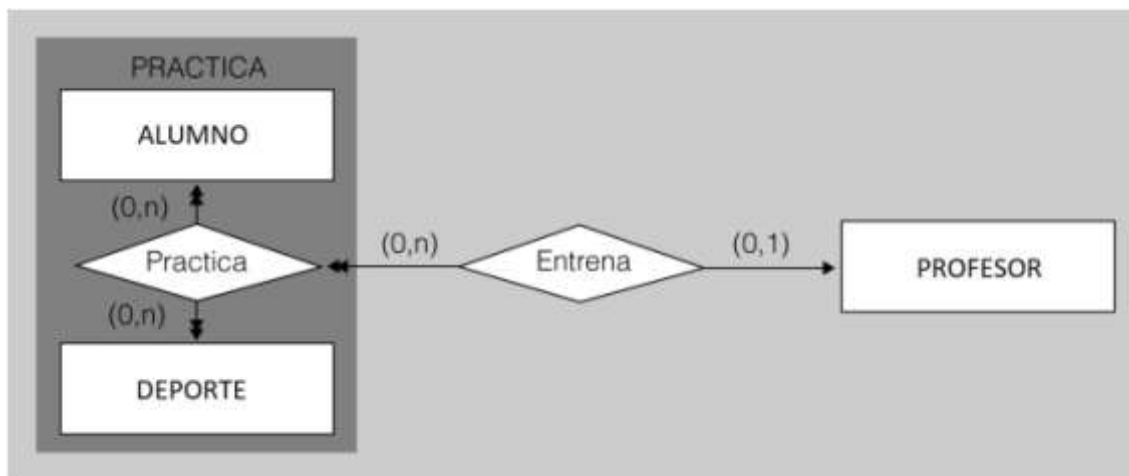


Figura 29 Ejemplo de agregación de entidades sin redundancia

La técnica de las agregaciones engloba la de las entidades débiles, pero aún resulta más potente: siempre que utilizamos una entidad débil, la podremos sustituir por una agregación, pero no al revés. Ahora bien, hay que mantener las entidades débiles en el modelo ER que, aunque resultan menos complejas que las agregaciones, normalmente son suficientes para modelizar la mayoría de las situaciones que se producen en el mundo real.

Ejemplo de sustitución de una entidad débil por una agregación

Recuperamos la entidad débil ASIGNATURA. Ahora imaginemos que, para establecer un cierto control en materia de coordinación pedagógica, se necesita saber quién es el profesor responsable de realizar la programación didáctica de cada asignatura.

Entre PROFESOR y ASIGNATURA se puede establecer una Relación binaria de cardinalidad 1:N para representar este hecho, que se llama, por ejemplo, Programa.

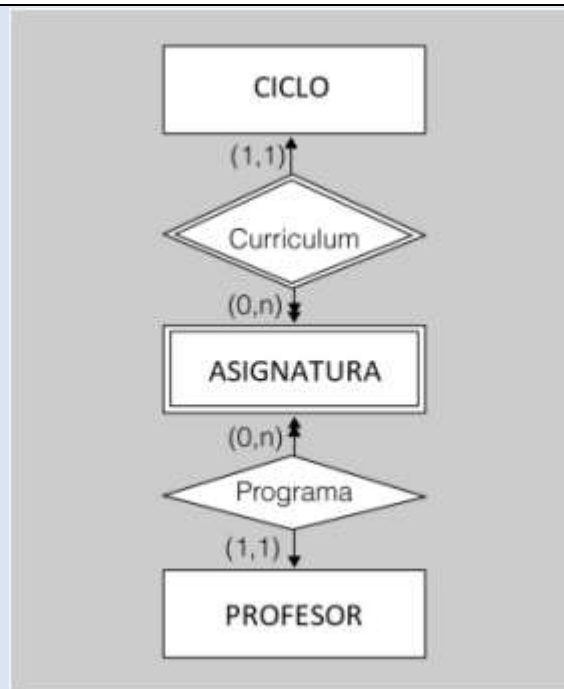


Figura 30

Pero, alternativamente, podríamos modelizar este dato convirtiendo ASIGNATURA en una agregación.

Para ello, en primer lugar deberíamos considerar una nueva entidad, y llamarla por ejemplo CÓDIGO, que almacenara simplemente códigos de asignatura (como C1, C2, C3, etc.).

A continuación, deberíamos establecer, por un lado, una Relación binaria de cardinalidad NM entre CICLO y CÓDIGO, y llamarla Asignatura. Y, por otro lado, también deberíamos obtener una agregación de la Relación entre CICLO y CÓDIGO (es decir, Asignatura).

Finalmente, deberíamos relacionar la agregación resultante con la entidad PROFESOR, con una sencilla Relación binaria con cardinalidad 1:N, llamada Programa (figura .13).

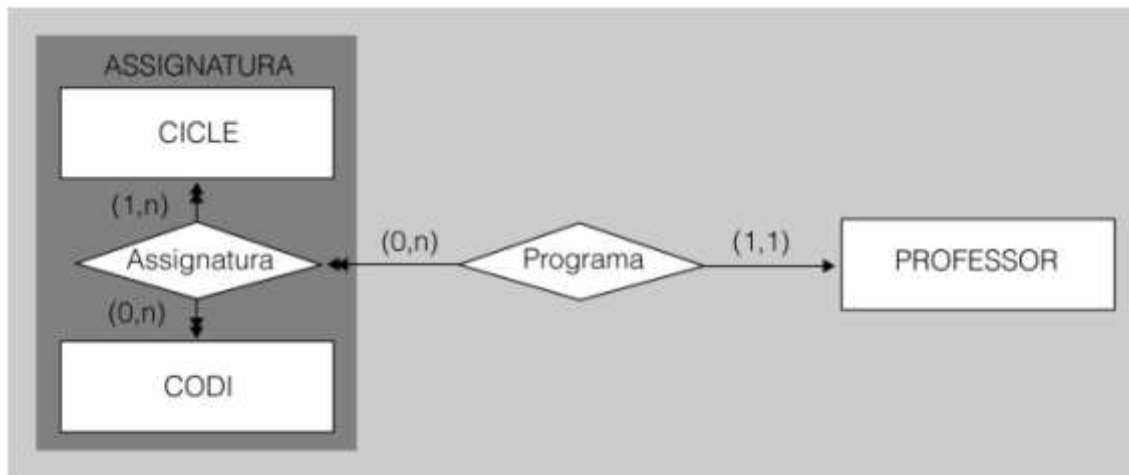


Figura 31 Ejemplo de sustitución de una entidad débil por una agregación (resultado)

Notación

Las agregaciones de entidades se representan incluyendo dentro de un recuadro todas las entidades que participan en una Relación determinada.

Las agregaciones también se representan frecuentemente incluyendo, dentro de un recuadro, sólo el rombo de la Relación de la cual provienen las entidades implicadas.

Desde una Relación, se puede hacer llegar una flecha (de punta sencilla o doble, con el fin de expresar la cardinalidad 1 o N, respectivamente) hasta el rombo incluido dentro del recuadro que indica la existencia de una agregación (o bien hasta el mismo recuadro, exactamente igual que si se tratara de una simple entidad).

Toda agregación debe tener el mismo nombre que la Relación sobre la que se define.

6. EJEMPLO: BD DE UN INSTITUTO DE FORMACIÓN PROFESIONAL

Ahora desarrollamos un ejemplo de diseño conceptual de BD, correspondiente a un instituto de formación profesional, para ilustrar por separado los diferentes conceptos y su respectiva modelización. Se trata de diseñar una BD para gestionar el personal del instituto (compuesto por los profesores, y los trabajadores de administración y servicios) y su alumnado, además de los estudios impartidos.

Las Descripcionnes siguientes resumen los requisitos de los usuarios de la futura BD:

Los requisitos para diseñar una BD ...

... de un instituto real seguramente serían mucho más numerosos que los que hemos expuesto aquí, los que están seleccionados con fines educativos, y no necesariamente en función de su importancia o frecuencia en el mundo real.

- Las personas que forman parte de nuestra comunidad educativa se identifican mediante el DNI (o documento equivalente, como la tarjeta de residencia).
- También queremos conocer, de estas personas, el nombre y apellidos, la dirección, uno (y sólo uno, de momento) teléfono de contacto, y la fecha de nacimiento.
- Además, debemos tener registrada la localidad de residencia, teniendo en cuenta que la BD debe poder almacenar, para otros usos, localidades donde no viva nadie.
- Toda persona de la comunidad educativa pertenece, como mínimo, a uno de los tres subtipos siguientes:
 - profesores
 - alumnos

- Personal de administración y servicios
 - Las personas sólo pueden mantener un tipo de relación laboral con el centro educativo. Por lo tanto, los profesores no pueden pertenecer simultáneamente al colectivo del personal de administración y servicios.
 - Tampoco está permitido que los profesores se matriculen en el centro de trabajo en ninguno de los estudios impartidos. Por lo tanto, los profesores no pueden considerarse simultáneamente alumnos.
 - En cambio, sí está permitido a los integrantes del colectivo de administración y servicios que se matriculen, fuera de su horario laboral, en alguno de los estudios impartidos. Por lo tanto, el personal de administración y servicios puede pertenecer simultáneamente a la categoría de alumno.
 - Debemos tener constancia del sueldo de los profesores y del personal de administración y servicios.
 - Organizativamente, todo profesor está asignado a un solo departamento. Y cada departamento tiene asignado uno de sus profesores como coordinador.
 - Todo profesor tiene reconocida una especialidad determinada (o más de una). Pero internamente, la BD del instituto sólo necesita registrar cuáles de los profesores que tiene asignados el centro pertenecen a las especialidades de informática o de administración, con el fin de asignarles tareas específicas adicionales a las docentes que les son propias.
 - De cada informático, queremos saber las especialidades profesionales, cuando las haya, tanto en el ámbito del hardware como también en el del software.
 - De cada administrativo, queremos conocer la titulación académica y la especialidad profesional, si tiene.
- Los alumnos pueden practicar algunos deportes en las instalaciones del centro. Y, incluso, pueden disponer de algunos profesores como entrenadores personales, que se han ofrecido voluntariamente para realizar esta tarea.
- Como es lógico, al tratarse de un centro de formación profesional, el instituto de nuestro ejemplo ofrece diferentes estudios estructurados en ciclos formativos, y cada ciclo formativo tiene sus propias asignaturas. Es, pues, codificar las asignaturas de la misma manera que se hace en el currículo oficial del ciclo formativo al que pertenecen. El problema es que estos códigos se repiten para todos los ciclos formativos, ya que la codificación siempre consiste en una C (por ser la inicial de la palabra código) seguida de un número entero (C1, C2, C3, y así sucesivamente).
- Dentro de un mismo ciclo formativo, se puede exigir a los alumnos que, para matricularse en algunas asignaturas, hayan superado alguna asignatura (o más de una).
- Por otro lado, siempre hay un profesor encargado de realizar la programación didáctica de cada asignatura. Un mismo profesor, pero, puede encargarse de la programación didáctica de más de una asignatura.
- Todos los alumnos del centro tienen un compañero que actúa como delegado en el ámbito de una asignatura y se encarga, por ejemplo, de distribuir los materiales o las baterías de ejercicios. Un mismo alumno puede actuar como delegado en el ámbito de más de una asignatura. Pero cada alumno sólo tendrá un delegado en cada asignatura en que esté matriculado.
- Finalmente, la BD debe recoger a qué asignaturas está matriculado cada alumno en cada curso académico, y la nota final obtenida.

La figura 14 muestra un diagrama ER que cumple los requisitos mencionados anteriormente.

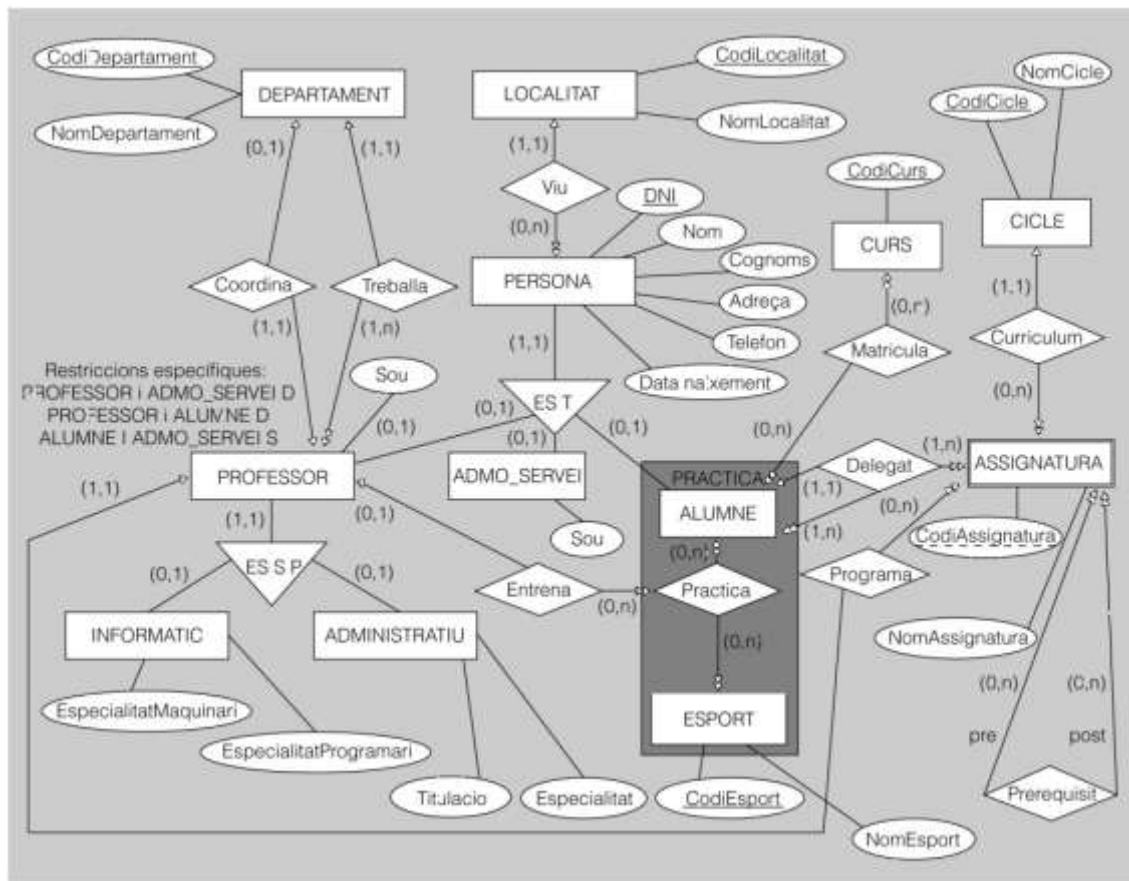


Figura 32 Ejemplo: BD de un instituto de formación profesional

A continuación, se muestra una lista de todas las entidades que aparecen en el diagrama, acompañadas de los respectivos atributos (subrayados si forman parte de una clave primaria).

- DEPARTAMENTO
 - CódigoDepartamento, NombreDepartamento
- LOCALIDAD
 - CódigoLocalidad, NombreLocalidad
- PERSONA
 - DNI, Nombre, Apellidos, Dirección, Teléfono, FechaNacimiento
- PROFESOR (subclase de PERSONA)
 - DNI, Sueldo
- ADMINISTRATIVO (subclase de PERSONA)
 - DNI, Sueldo
- ALUMNO (subclase de PERSONA)
 - DNI
- INFORMATICO (subclase de PROFESOR)
 - DNI, EspecialidadSistemas, EspecialidadSoftware
- ADMINISTRATIVO (subclase de PROFESOR)
 - DNI, titulación, Especialidad
- DEPORTE

- *CodigoDeporte, Deporte*
- CURSO
 - *CodigoCurso*
- CICLO
 - *CodigoCiclo, NombreCiclo*
- ASIGNATURA (entidad débil: *CodigoAsignatura* la identifica parcialmente, y necesita el código del ciclo correspondiente para identificarse completamente)
 - *CodigoAsignatura, NombreAsignatura*

Finalmente, cabe comentar algunos de los aspectos más complejos de este modelo, proporcionado a modo de ejemplo:

- Las subclases en que se especializa PROFESOR (INFORMATICO y ADMINISTRATIVO) son solapadas (S) entre ellas, y además parciales (P):
 - *Son solapadas para que las instancias de la superclase pueden pertenecer simultáneamente a ambas categorías.*
 - *Son parciales, porque no todas las instancias de la superclase deben pertenecer necesariamente a alguna de las dos categorías.*
- Las subclases que dan lugar a la generalización de PERSONA (PROFESOR, ADMINISTRATIVO y ALUMNO) son totales, para que toda instancia de la superclase debe pertenecer simultáneamente a alguna de las tres subclases mencionadas. Ahora bien, respecto a si pueden pertenecer simultáneamente a diferentes subclases o no, tienen restricciones específicas, y las combinan de dos en dos. Esta particularidad está especificada textualmente dentro del diagrama:
 - *PROFESOR y ADMINISTRATIVO: las entidades son disjuntas entre ellas, para que las instancias respectivas no pueden pertenecer al mismo tiempo a las dos.*
 - *PROFESOR y ALUMNO: se da la misma circunstancia que en el caso anterior.*
 - *ALUMNO y ADMINISTRATIVO: las entidades son solapadas entre ellas, para que las instancias respectivas sí pueden pertenecer al mismo tiempo a las dos.*
- Entre DEPARTAMENTO y PROFESOR hay dos relaciones porque sirven para modelizar dos realidades diferentes: la coordinación del departamento por parte de un profesor (con cardinalidad 1 : 1), y el hecho de que una pluralidad de profesores adscritos al mismo (con cardinalidad 1 : N).
- La localidad de residencia de las personas se ha implementado mediante una entidad independiente, y no como un simple atributo de la entidad PERSONA. De este modo, se evitarán redundancias, porque cada localidad sólo se registrará una vez dentro de la BD, aunque después relacionará con todas las instancias de PERSONA que sea necesario.
- Se ha optado por establecer una agregación a partir de la Relación Practica, para permitir establecer otra Relación (Entrena) entre ésta y PROFESOR, que evita la redundancia de datos que habría si se hubiera utilizado una Relación ternaria entre ALUMNO, DEPORTE y PROFESOR, ya que contendría todas las combinaciones de la Relación entre ALUMNO y DEPORTE. Y no se podría implementar simplemente una ternaria entre ALUMNO, DEPORTE y PROFESOR, y suprimir la binaria mencionada, para que los alumnos también pueden practicar los deportes por libre, sin ningún profesor que los entrene.
- Para representar la figura del alumno delegado de asignatura, se ha recurrido a una Relación recursiva ternaria, ya que es necesario relacionar cada alumno que actúa como delegado con sus alumnos representados y, además, con la asignatura de la que se trate en cada caso.
- Para representar los prerequisites de matriculación, hemos añadido otra Relación recursiva, en este caso binaria, que sirve para asociar las asignaturas entre ellas cuando es necesario.
- Fijémonos en que la Relación ternaria Matrícula, entre CURSO, ALUMNO y ASIGNATURA, con cardinalidad M: N: P, tiene un atributo propio con el fin de almacenar la nota final de cada alumno.

7. EL 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.

7.1. 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 le 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.

7.1.1. 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.

7.1.2. 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.

7.1.3. 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.

7.1.4. 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*.

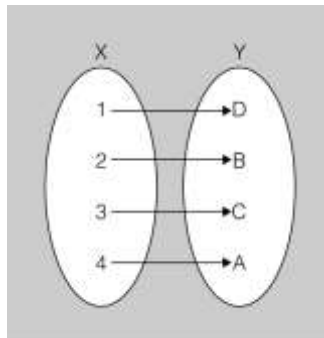
Una **clave foránea** está constituida por un atributo, o por un conjunto de atributos, del esquema de una relación, que sirve para relacionar sus tuplas con los tuplas de otra relación de la base de datos (o con los tuplas de sí misma, en algunos casos).

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, DNIDelegdo 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 **I**.

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)

7.1.5. 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

7.2. 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

7.2.1. 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 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 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

7.2.2. 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

7.2.3. 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.

7.2.4. 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.

7.3. TRADUCCIÓN DEL MODELO ENTIDAD-RELACIÓN 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 Entidad-Relación (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 narias.

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.

7.3.1. 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)

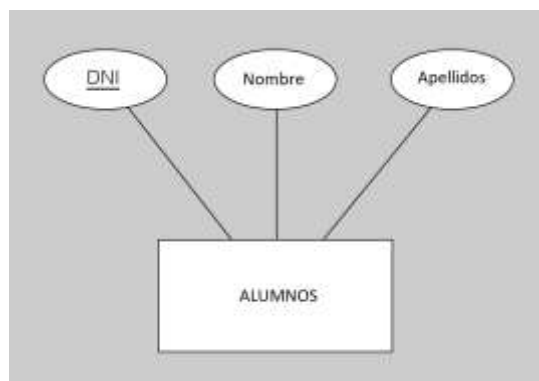


Figura 33

7.3.2. 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 {CódigoDepartamento} 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.

b.

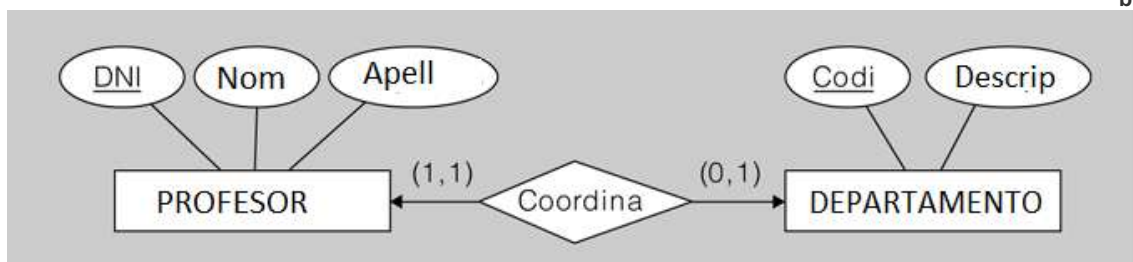


Figura 34 Interrelación binaria con conectividad 11

- 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.

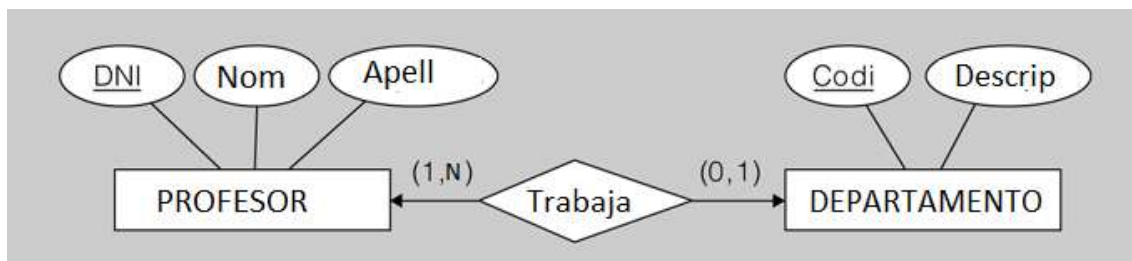


Figura 35 Interrelación binaria con conectividad 1N

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.

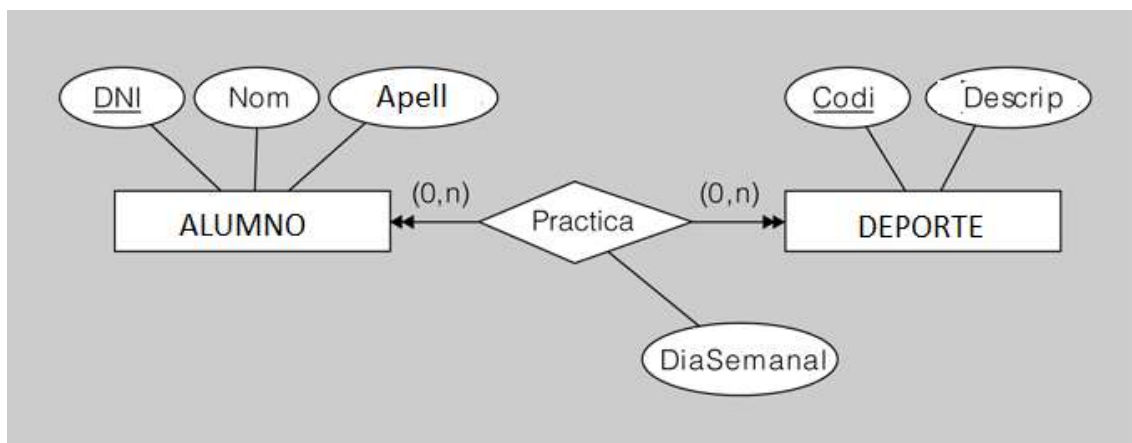


Figura 36 Interrelación binaria con conectividad M-N

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* , Descripcion)

CURSO(*Código*)

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

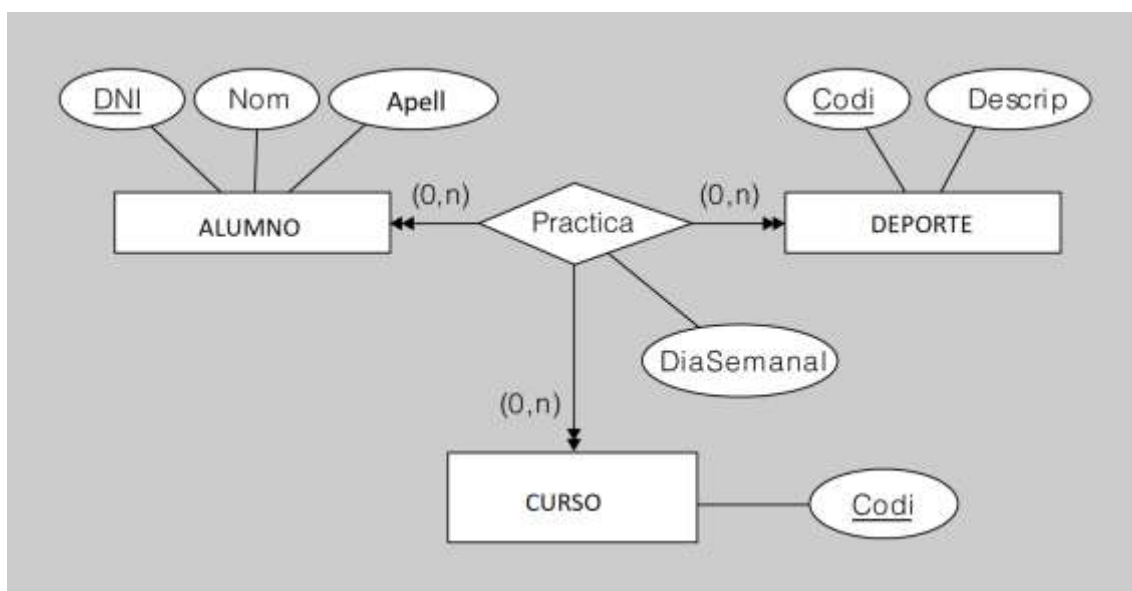


Figura 37 Interrelación ternaria con conectividad MNP

- 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.

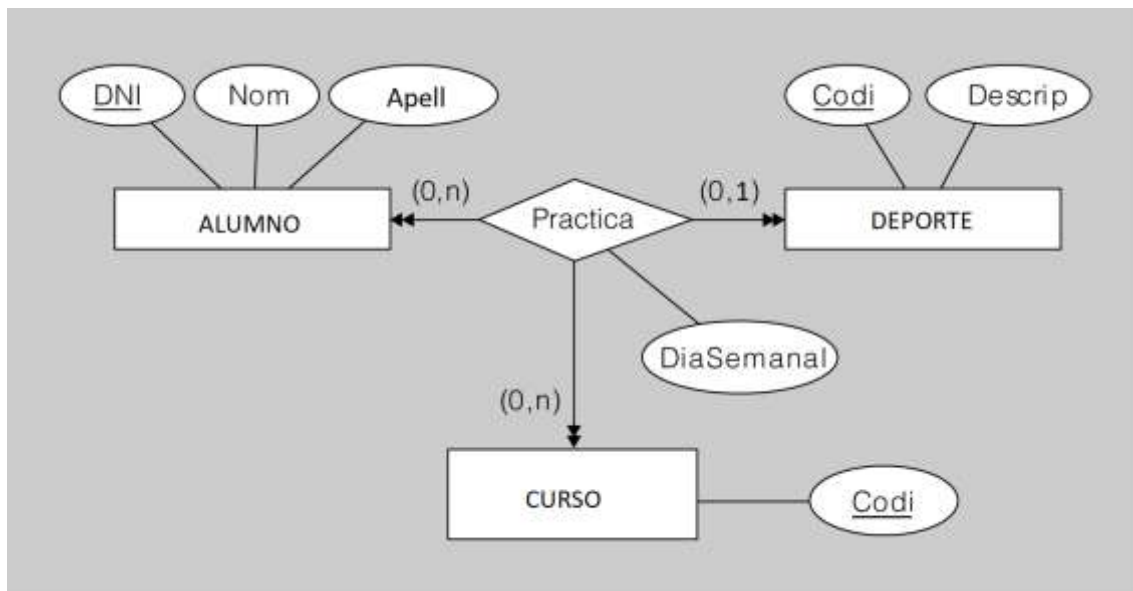


Figura 38 Interrelación ternaria con conectividad 1MN

- 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.

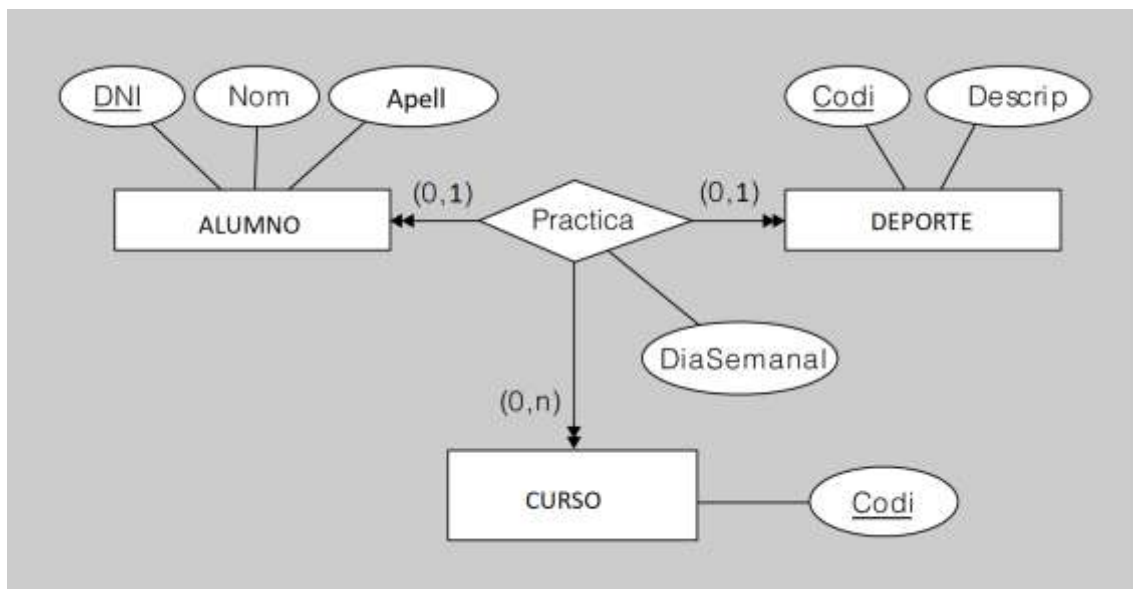


Figura 39 Interrelación ternaria con conectividad 11N

- 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.

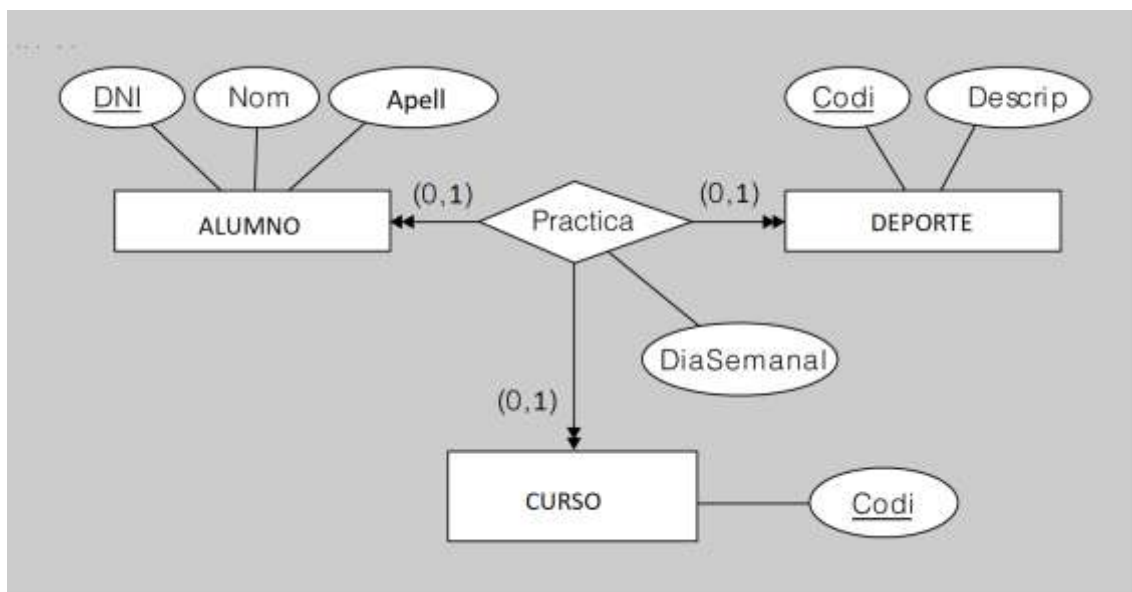


Figura 40 Interrelación ternaria con conectividad 1-1-1

- 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.

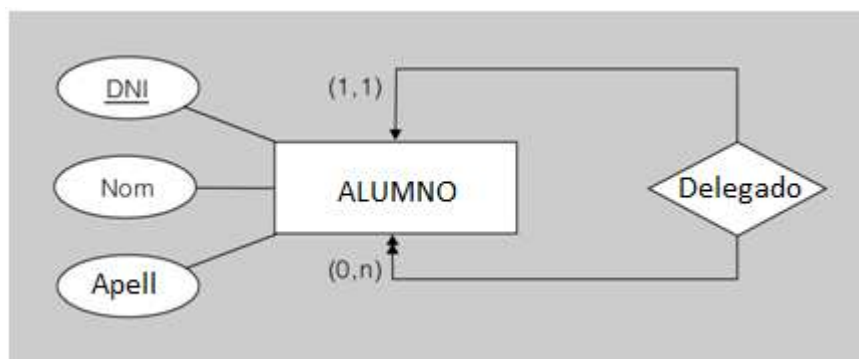


Figura 41 Interrelación recursiva binaria con conectividad 1-N

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 , CódigoPrerrequisito) ON {CodiAsignatura}

REFERENCIA ASIGNATURA (Código) y {CodiPrerrequisito} REFERENCIA ASIGNATURA (Código)

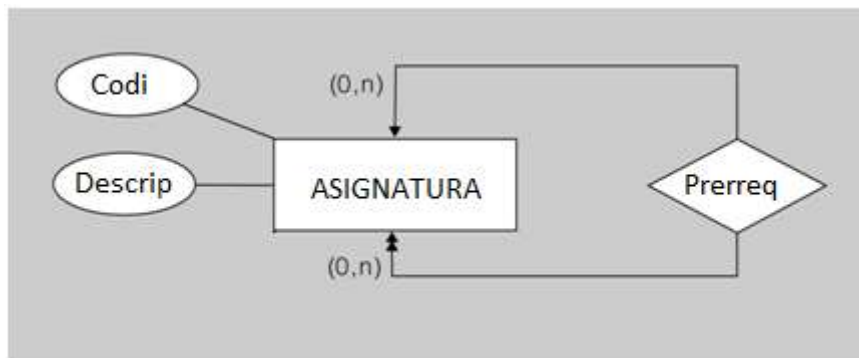


Figura 42 Interrelación recursiva binaria con conectividad M-N

- 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.

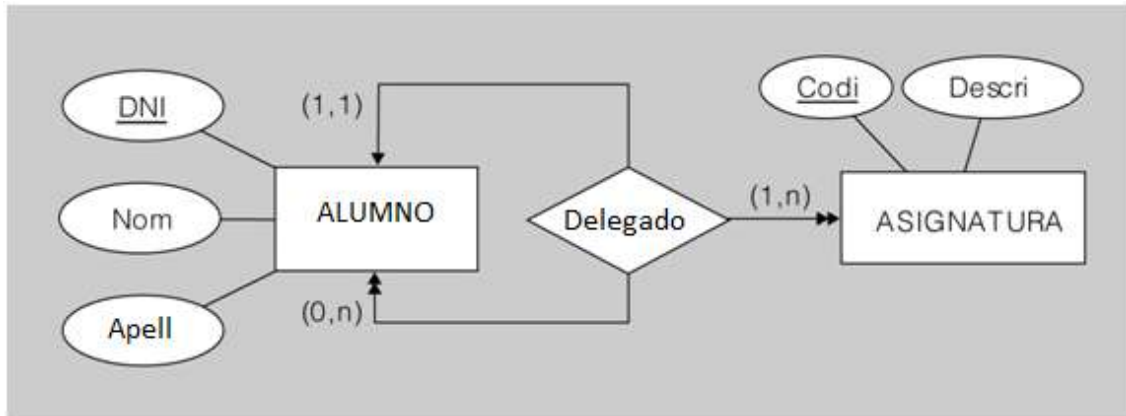


Figura 43 Interrelación recursiva naria

7.3.3. 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

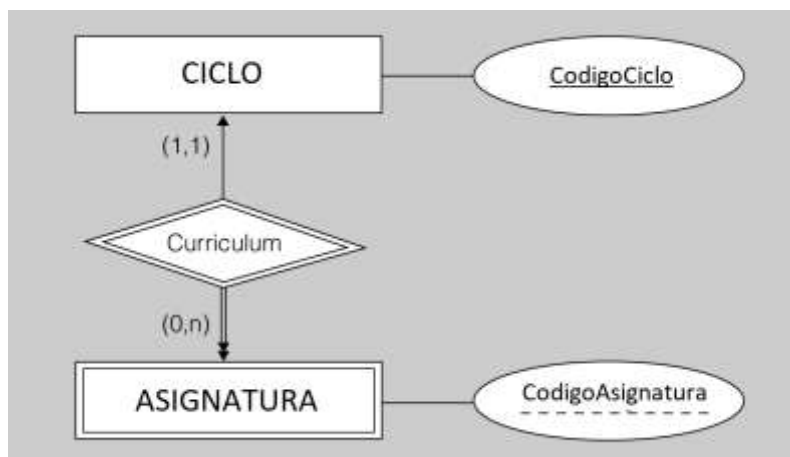


Figura 44 Entidad Débil

7.3.4. 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

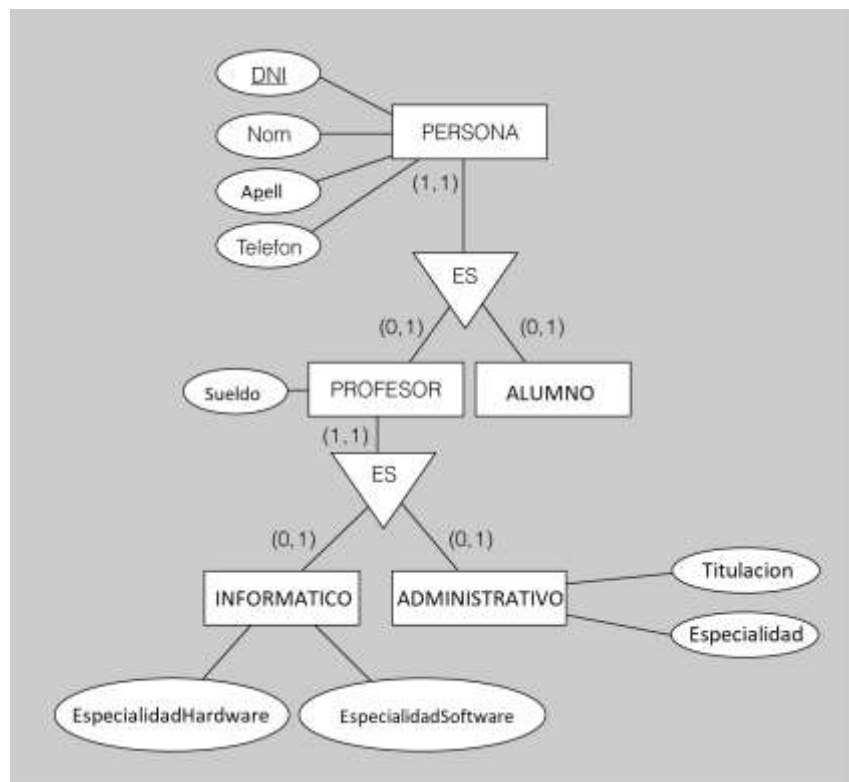


Figura 45 Generalización y especialización

7.3.5. 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.

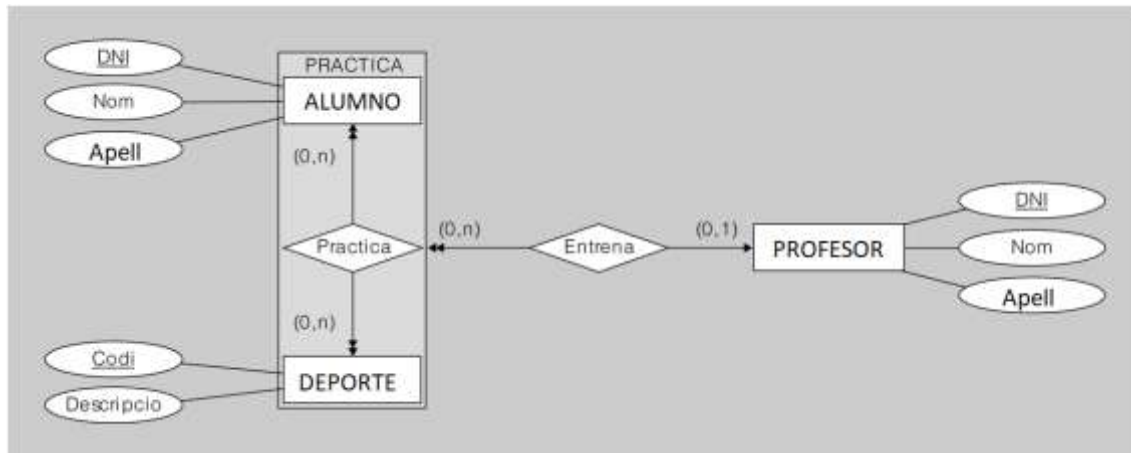


Figura 46 Entidad asociativa

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, {CodidoDeporte} 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.