CAPÍTULO 2. FASES DEL DISEÑO DE UNA BASE DE DATOS

1. Fase de Análisis de Requisitos: Modelo Entidad Interrelación (E/R)

1.1. Introducción

Los datos constituyen en la actualidad el arma más poderosa de cualquier organización o empresa. Una buena gestión de los datos puede influir de manera más que notable en los beneficios de cualquier organización. Pongámonos en el caso de una entidad bancaria y pensemos en los miles de clientes con cuyos datos se realizan operaciones diarias; la mala utilización de los mismos puede traer consigo pérdidas enormes para la empresa. En ocasiones esta mala utilización puede ser debida a la falta de formación de los empleados, pero muchas veces es ocasionada por un mal diseño del sistema de información o base de datos que gestiona los datos.

Hoy en día todas las empresas cuentan con herramientas informáticas de creación de bases de datos; entonces, ¿por qué se producen fallos?. La respuesta no está en las herramientas en sí, sino, y reincidiendo en el tema, en cómo se diseña la base de datos. Cada herramienta dispone de sus propios utensilios de diseño, pero todos ellos se basan en los mismos conceptos teóricos, conceptos que si se desconocen no pueden ser aplicados.

Por lo dicho anteriormente parece, si no completamente necesario, sí al menos muy conveniente, la utilización de un modelo de datos que permita diseñar bases de datos a nivel conceptual (y por tanto muy cercana al usuario) y por supuesto la formación de personal cualificado en este campo.

El modelo Entidad/Interrelación (E/R) es un modelo conceptual que ha demostrado ser muy válido para cumplir con este objetivo, pues está a un nivel de abstracción lo suficientemente elevado como para poder diseñar cualquier base de datos con independencia de la máquina en la que se implemente. Además, en la actualidad disponemos en el mercado de una amplia gama de herramientas que automatizan en gran parte las tareas del diseño y que toman como base este modelo de datos.

El modelo E/R fue propuesto por Peter Chen en 1976. Desde entonces muchos autores se han interesado por él, estudiándolo y ampliándolo, consiguiendo así diversas variantes del modelo (distintas formas de representación de los objetos), pero todas ellas parten del mismo concepto: el conocimiento del *mundo real* que se desea representar a través de un análisis de los requisitos o especificaciones del problema.

En la realización del esquema o diseño conceptual de cualquier base de datos es fundamental el conocimiento del problema a modelar y es en este conocimiento donde representan un papel primordial los usuarios finales del sistema, pues es en esta primera etapa de modelización en la que el diseñador de la base de datos debe hacer tantas entrevistas como sean necesarias con los usuarios para conseguir clarificar todas las especificaciones del problema. Una vez clarificados los objetivos y las necesidades se deberá pasar al diseño propiamente dicho de la base de datos.

El modelo E/R, como todos los modelos, consiste en un conjunto de conceptos, reglas y notaciones que permiten formalizar la semántica del mundo real que se pretende modelar (también denominada Universo del Discurso) en una representación gráfica o diagrama que denominamos esquema de la Base de Datos.

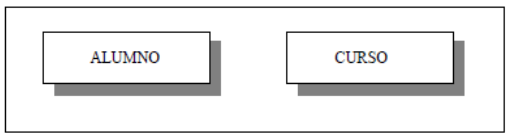
1.2. Elementos básicos del Modelo E/R

Los elementos u objetos básicos del modelo E/R son cuatro: entidades, interrelaciones, atributos y dominios. A continuación se explican cada uno de ellos.

Entidades

Las **entidades**, también llamadas tipos de entidad, representan conjuntos de elementos con existencia propia y que se caracterizan por las mismas propiedades. Generalmente son personas, cosas, lugares,..., es decir, conceptos sobre los que necesitamos guardar información y distinguibles de los demás objetos. Su representación gráfica se hace por medio de un rectángulo dentro del cual se escribe el nombre de la entidad en mayúsculas (generalmente un sustantivo).

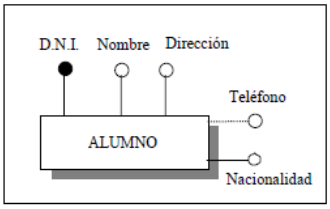
Por ejemplo, si queremos diseñar una base de datos para gestionar todos los alumnos de los cursos Mentor, entre los tipos de entidad que deberíamos definir estarían ALUMNO y CURSO. El primero representaría el conjunto de todos los alumnos que se inscriben en los diferentes cursos, el segundo recogería todos los cursos ofertados por el aula Mentor.



Atributos

Todo tipo de entidad tiene unas características o cualidades propias que queremos recoger dentro de nuestro diseño. El modelo E/R define estas cualidades como **atributos**, así por ejemplo el *nombre del alumno, el teléfono*, etc., describen propiedades de cada uno de los miembros que pertenecen al tipo de entidad ALUMNO. Estas propiedades no tienen existencia propia, es decir, sólo tienen sentido en el esquema de la Base de Datos en tanto en cuanto aparecen formando parte de una entidad o, como veremos más adelante, de otro de los elementos del modelo E/R, de una interrelación.

Supongamos que de cada alumno queremos la información referente a su D.N.I., Nombre, Dirección, Teléfono y Nacionalidad.



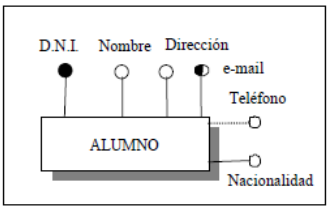
Es importante destacar que un mismo concepto no tiene por qué representarse siempre de la misma forma (por ejemplo, como una entidad o como un atributo). Así, si estuviéramos modelando una Base de Datos para una tienda de ropa, probablemente tendríamos una entidad denominada PRENDA y uno de sus atributos podría ser *Color* (roja, negra, etc.). Sin embargo, si estuviéramos hablando de una Base de Datos para gestionar la información de un taller de vehículos dedicado a trabajos de chapa y pintura, el concepto de color puede tener tal importancia que pase a ser una entidad COLOR, pues tiene existencia propia y un conjunto de propiedades (*código de color*, *textura*, *tipo de mezcla*, etc.).

Tipos de atributos

En el ejemplo aparece el atributo D.N.I. con un círculo negro, este tipo de atributo se denomina **identificador principal (IP)** y lo que indica es que el atributo o propiedad DNI es único para cada ejemplar del tipo entidad ALUMNO.

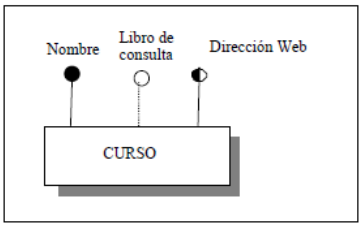
Para poder distinguir una ejemplar de otra, dentro de un mismo tipo de entidad, el modelo E/R obliga a que cada vez que definimos un tipo de entidad se defina un atributo que identifique cada ejemplar, es decir, un IP. Por lo tanto en todos los tipos de entidad tiene que aparecer de forma obligatoria una característica que identifique de forma única cada uno de los ejemplares. Esta es la representación que nos proporciona el modelo E/R para distinguir este tipo de atributo del resto de atributos que componen el tipo de entidad. En un tipo de entidad sólo puede aparecer un **identificador principal**, pero pueden existir distintos atributos que también identifiquen los ejemplares de esta; este tipo de atributos se denominan **Identificadores Alternativos (IA)**.

Veamos un ejemplo, supongamos que queremos añadir para el tipo de entidad ALUMNO, la dirección de correo electrónico que este posee, sabiendo que es única para cada uno de los alumnos. El atributo *e-mail* sería un identificador alternativo y como vemos en la siguiente figura se representa con un círculo mitad negro mitad blanco, indicando que su valor es único para cada ejemplar del tipo de entidad ALUMNO.



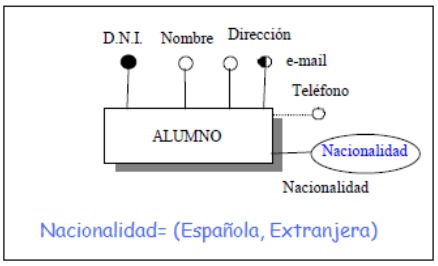
El atributo *Teléfono* aparece representado con una línea de puntos lo que significa que estamos ante un **Atributo Opcional** que nos informa de que existen alumnos que puede que no tengan número de teléfono o que al fin y al cabo es un atributo cuyo valor no es demasiado importante y por eso no lo ponemos como obligatorio. Por tanto, cuando los valores de un atributo van a ser desconocidos o por alguna otra causa no van a tener valor se denominan **Atributos Opcionales**.

Supongamos que para el tipo de entidad CURSO es importante recoger las siguientes propiedades: nombre, libro de consulta y dirección Web. De estas tres características de CURSO elegiremos como identificador principal el *nombre*, ya que cada curso tiene un nombre distinto, la *dirección Web* sería un identificador alternativo porque toma valores únicos para cada curso y *libro de consulta* sería un atributo opcional ya que permitimos que haya cursos que no tengan o que desconozcamos su libro de referencia.



Dominios

Supongamos que el atributo nacionalidad, sólo puede tomar los valores "española" o "extranjera". Para los conjuntos de valores sobre los que se definen los atributos utilizaremos un objeto del modelo E/R denominado **Dominio**. Un dominio se define por un nombre y un conjunto de valores.

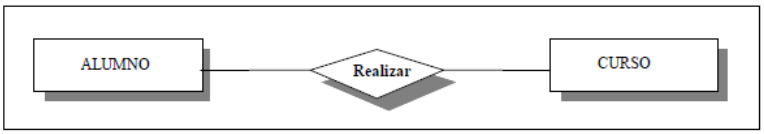


En general los dominios no se suelen representar en el modelo por problemas de espacio, pero para tener constancia de los valores que puede tomar un atributo se suele anotar después de la representación gráfica una representación textual.

Interrelaciones

Las interrelaciones representan asociaciones del mundo real entre una o más entidades. Por ejemplo, presentábamos los alumnos y los cursos del Mentor como entidades sin ningún tipo de relación, pero para poder expresar que un alumno está matriculado en distintos cursos y que en un curso se pueden matricular alumnos necesitamos una **Interrelación** que nos muestre la asociación existente entre ellos. Por lo tanto, vemos la necesidad de poder representar este concepto ya que aparece continuamente en el mundo real; algunos ejemplos son: "las sucursales de una entidad bancaria están relacionadas con sus clientes", "las editoriales se relacionan con los libros que publican", "los tutores de los cursos Mentor tienen asignados una serie de alumnos", etc.

Gráficamente las interrelaciones se representan mediante un rombo unido a los tipos de entidad mediante líneas; dentro del rombo se escribe el nombre de la interrelación en minúsculas, que en general, suele coincidir con un verbo en infinitivo.



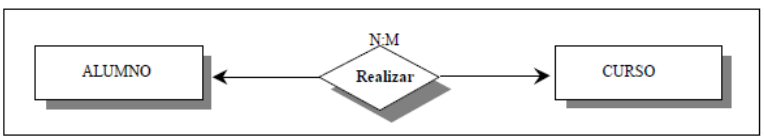
No todas las relaciones o asociaciones son iguales, en general se dividen en relaciones que denominamos de *uno a muchos,* como por ejemplo la que presentamos a continuación: "una sucursal es únicamente de una entidad bancaria (*uno*) pero una entidad bancaria posee varias sucursales (*muchos*)". También existen las relaciones *muchos a muchos*, como por ejemplo "un curso Mentor tiene asociados tutores (*muchos*) y los tutores pueden tutorar distintos cursos Mentor (*muchos*)".

Para poder recoger estas características que nos distinguen unas relaciones de otras, que nos permite, además, recoger más información acerca del problema que estamos modelando, vamos a introducir los siguientes propiedades de una interrelación: grado, tipo de correspondencia y cardinalidad.

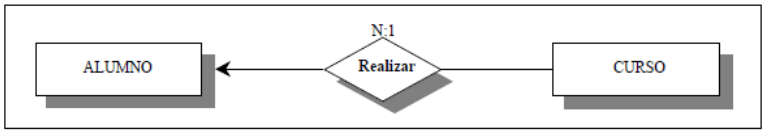
El grado de una interrelación es el número de entidades que intervienen en ella, debe ser como mínimo dos, es decir, el número de entidades que intervienen en una interrelación debe ser de al menos dos; existe un caso especial en el que sólo participa una entidad en la interrelación aunque de dos formas distintas (es lo que se denomina interrelación reflexiva, como se verá después).

En la figura anterior se representa una interrelación binaria, denominada así por tratarse de una interrelación entre dos tipos de entidad. De la misma forma, cuando el grado es tres se habla de interrelaciones ternarias y, en general, de interrelaciones n_arias cuando el grado es n. El tipo de interrelaciones que aparece de forma habitual en el modelado de una Base de Datos es la interrelación binaria y a partir de ahora nos centraremos solo en este tipo de interrelaciones.

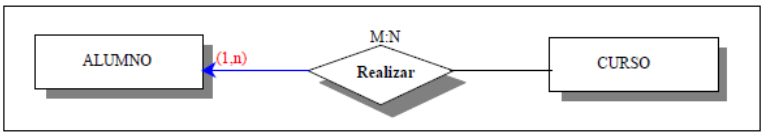
El Tipo de correspondencia de una interrelación binaria se define como el número máximo de ejemplares de un tipo de entidad que pueden estar asociados con un ejemplar del otro tipo de entidad. Su representación gráfica se hace por medio de un par X:Y colocado sobre el rombo de la interrelación, donde X e Y representan los ejemplares asociadas de los tipos de entidad en estudio. En nuestro ejemplo, en principio, el número de cursos a los que un alumno puede optar es ilimitado y el de alumnos que realizan un curso también, por tanto la correspondencia sería N:M o muchos a muchos.



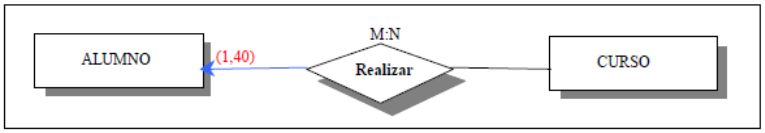
Si, por el contrario, en las especificaciones del problema se nos dijera que cada alumno solo puede matricularse de un curso, el tipo de correspondencia entre ALUMNO y CURSO cambiaría, sería 1:N o uno a muchos.



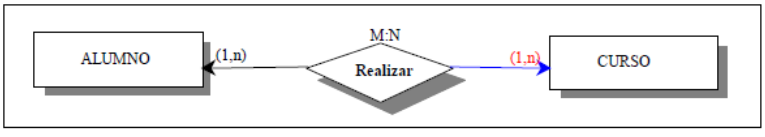
La cardinalidad de un tipo de entidad que interviene en una interrelación binaria se define como el número mínimo y el número máximo de ejemplares de un tipo que pueden relacionarse con un elemento de otro tipo de entidad. Para representar las cardinalidades utilizamos un par (x, y) situado sobre la línea que une el tipo de entidad con la interrelación, donde x indica el número mínimo e y el número máximo. Además, y cuando la cardinalidad máxima es n, se dibuja una punta de flecha hacia la entidad correspondiente. En el ejemplo que nos ocupa y suponiendo que no se establece ninguna restricción adicional, el número mínimo de alumnos que pueden matricularse en un curso es uno (no tendría sentido un curso con 0 matriculados), y el número máximo n (número ilimitado), por tanto la cardinalidad del tipo de entidad ALUMNO es (1,n).



La interpretación de la interrelación Realizar sería "un curso Mentor es realizado como mínimo por un alumno y como máximo n". Si tuviéramos limitación en la matriculación de los alumnos en un curso, por ejemplo, los cursos Mentor como máximo admiten 40 alumnos, lo representaríamos de la siguiente forma:



De la misma manera, el número mínimo de cursos que puede realizar un alumno es uno y el máximo n, es decir, la cardinalidad de CURSO es (1,n) y por tanto tendríamos que representar la punta de flecha hacía la entidad CURSO y encima de esta línea la cardinalidad.



Las cardinalidades mínimas y máximas son, como se puede apreciar, una extensión del tipo de correspondencia y nos dan más información referente al tipo de interrelación que estamos representando.

Veamos otro ejemplo con la relación que existe entre los empleados de una empresa y el departamento en el que trabajan. Sabemos que un empleado trabaja en un departamento y que a cada departamento se le asigna al menos un empleado. De cada empleado se desea la siguiente información: un código de empleado (número que le identifica), DNI, nombre completo, dirección, teléfono y número de afiliación de la seguridad social. Para los departamentos necesitamos un nombre, único para cada uno de ellos, una localización y un número de teléfono.

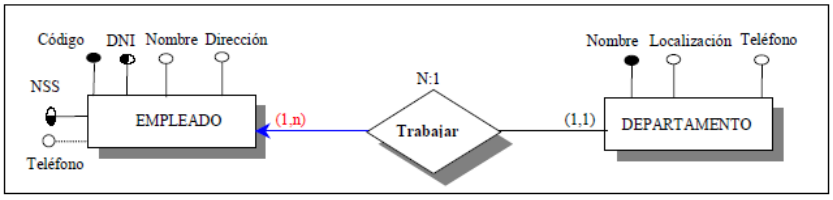
¿Cuál sería su diseño en el modelo E/R?.

Podemos detectar de forma clara que necesitamos dos entidades, EMPLEADO y DEPARTAMENTO, objetos que tienen existencia propia con determinadas características. Para la entidad EMPLEADO tenemos como identificador principal el código de empleado; el DNI, que es único para cada empleado será un atributo alternativo ya que hemos elegido el código como identificador principal por especificaciones del problema.

Los atributos *nombre* y *dirección* de EMPLEADO son obligatorios ya que dicha información la consideramos importante; por ejemplo, sin ellos no podríamos mandar la nómina o contactar por cualquier causa con los empleados de la empresa. El *teléfono* lo podemos considerar como un atributo opcional y, por último, el número de afiliación de la seguridad social (*NSS*) al tomar valores únicos para cada empleado lo consideraremos un atributo alternativo.

La interrelación que une las entidades representadas anteriormente, EMPLEADO y DEPARTAMENTO, es binaria ya que relaciona dos entidades; el tipo de correspondencia es 1:N o de uno a muchos, ya que un empleado está asignado a un departamento y a un departamento pertenecen varios empleados.

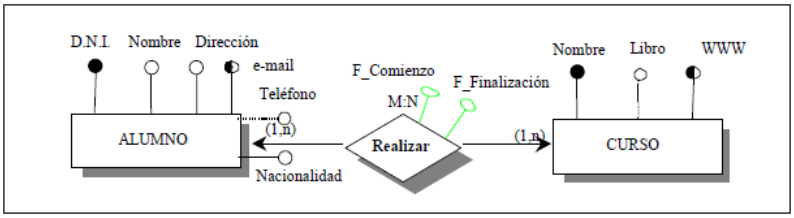
La interpretación de la interrelación Trabaja sería la siguiente: "Un empleado trabaja como mínimo y como máximo en un solo departamento (1,1)" y tendríamos una línea continua entre el rombo de la interrelación y la entidad DEPARTAMENTO para reflejar este hecho.



Atributos de una interrelación

Como ya se ha mencionado, los atributos no solo están referidos a los tipos de entidad. Las interrelaciones también pueden tener atributos propios, atributos cuyos valores tienen sentido

únicamente en el caso de que se establezca la relación entre los tipos de entidad que las une, como pueden ser las fechas de comienzo y de finalización de un curso, que no tienen sentido si dicho curso no es realizado por al menos un alumno.

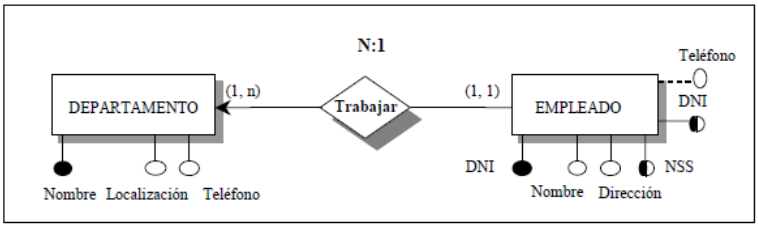


1.3. Extensiones del Modelo E/R

Posteriormente al modelo E/R propuesto por Chen se realizaron algunas extensiones para darle más riqueza semántica. Esto significa que se le han añadido nuevos conceptos para que el modelo se adapte mejor a la realidad que queremos modelar, es decir, recoja mayor semántica.

Vamos a introducir algunos de estos nuevos conceptos retomando el ejemplo visto en el apartado anterior sobre una empresa en el que habíamos representado la relación que existía entre los empleados y los departamentos de la empresa.

Supongamos que la empresa es un consorcio de distintas librerías especializadas en libros y revistas informáticas llamada INTERFAZ, y sabemos que los empleados de INTERFAZ están asignados a un departamento.



Entidades

En el apartado 2 se estudió que las entidades en un esquema E/R son los objetos principales sobre los que debe recogerse información y generalmente denotan personas, lugares, cosas o eventos de interés. En esta sección vamos a estudiar cómo las entidades pueden clasificarse por la fuerza de sus atributos identificadores.

Las entidades fuertes o regulares tienen existencia propia, es decir, poseen identificadores internos que determinan de manera única la existencia de sus ejemplares. Las entidades débiles son dependientes de otras entidades y pueden serlo por dos motivos: bien porque la existencia de sus ejemplares en la base de datos depende de una entidad fuerte bien porque sus

ejemplares requieran para su identificación de los atributos identificadores (algunas veces llamados atributos externos) de otra entidad.

Por ejemplo, los ejemplares correspondientes a los alumnos del MENTOR no dependen de ninguna otra entidad para existir en la base de datos; por ello la entidad ALUMNO es una entidad fuerte. Sin embargo, en el caso de una base de datos de una cadena hotelera podríamos tener el tipo de entidad HABITACIÓN dependiente del tipo de entidad HOTEL ya que para que existan ejemplares de HABITACIÓN es necesario que existan ejemplares de HOTEL. Una ejemplar de HABITACIÓN no tiene existencia por si misma porque siempre estará asociada a una ejemplar de HOTEL. Además, si se elimina un determinado ejemplar de la entidad HOTEL de la base de datos también deberán desaparecer los ejemplares de la entidad HABITACIÓN asociadas a él. La representación de una entidad débil difiere de la de una entidad regular pues el rectángulo de la entidad débil es de doble recuadro.

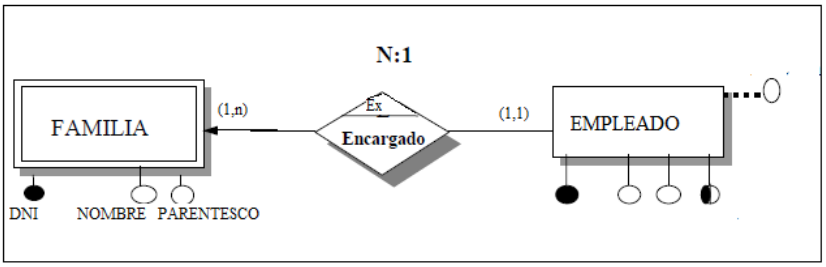
Interrelaciones binarias

La clasificación anterior entre entidades fuertes y débiles da lugar a dos tipos de interrelaciones según los tipos de entidades que asocian. Las interrelaciones regulares relacionan tipos de entidades regulares o fuertes. Las interrelaciones débiles relacionan un tipo de entidad regular y un tipo de entidad débil. Además, en las interrelaciones débiles podemos distinguir:

Dependencia en existencia.
Este tipo de interrelación refleja que los ejemplares del tipo de entidad débil que se relacionan con un determinado ejemplar del tipo de entidad regular dependen de él y, si éste desaparece, ellos también.

Supongamos que la empresa INTERFAZ necesita conocer los datos de los familiares que están a cargo de cada empleado de la empresa (cónyuge, hijos, etc.) para de esta manera apoyar a aquellos cuya carga familiar sea numerosa.

Para saber los familiares que dependen de cada empleado debemos crear un nuevo tipo de entidad, que denominaremos FAMILIAR, cuyos atributos podrían ser el DNI (como IP), el nombre completo y parentesco con el empleado. Como se puede observar, la existencia de un miembro de la familia depende plenamente de que ese miembro tenga a una persona de su familia trabajando en la empresa, o lo que es lo mismo que exista un ejemplar de EMPLEADO que esté relacionado con él; es decir, los familiares sólo existen en la base de datos si existe un empleado con el que se relacionen y si un determinado EMPLEADO se va de la empresa, entonces se eliminarán todas los ejemplares de FAMILIAR que dependan de él.

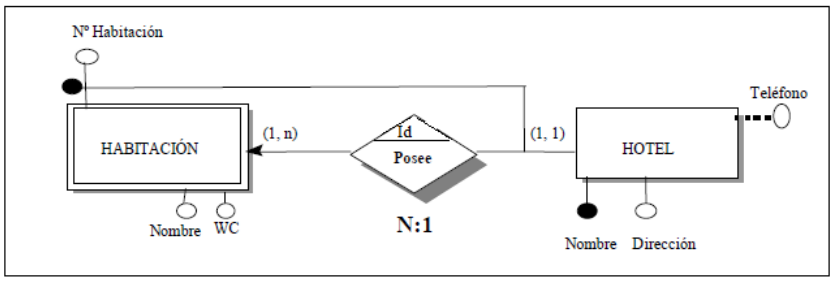


Dependencia en Identificación:
Este tipo de interrelación complementa a la anterior en que, además de que los ejemplares del tipo de entidad débil dependen de la existencia de un ejemplar de la entidad regular, también necesitan para su identificación el IP de la entidad regular. Así, veíamos anteriormente que la entidad HABITACIÓN era débil respecto al HOTEL al que pertenece. Si construimos las interrelación existente entre ambas entidades debemos pensar si el atributo "Nº de Habitación" de la entidad HABITACIÓN es suficiente para identificar cada ejemplar de esta (es decir, el número de habitación se repite para distintos hoteles).

Posibles soluciones:

a) La primera consiste en cambiar el IP, por ejemplo, poner el nombre de la habitación como IP; esto significa que los nombres de la habitación no pueden repetirse en los distintos hoteles y esto no es posible asegurarlo.

b) La segunda, y más razonable, consiste en crear una interrelación débil de dependencia en identificación, es decir, los ejemplares de la entidad débil requieren para su identificación de los atributos identificadores de la entidad fuerte. Así, cada ejemplar de HABITACIÓN está identificada por la concatenación de su número y del nombre del hotel en que se encuentra.

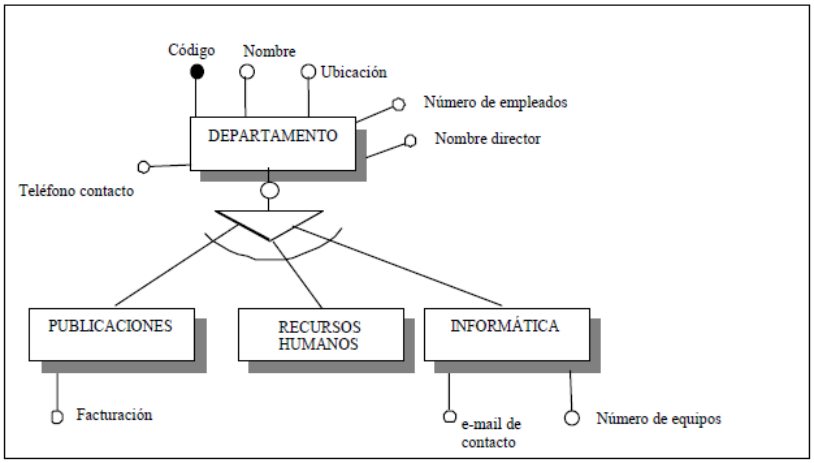


Interrelaciones jerárquicas

Otro tipo de interrelación es la denominada jerárquica que expresa la clasificación de un determinado tipo de entidad en uno o más tipos de entidad. Por ejemplo, supongamos que la empresa Interfaz tiene tres departamentos INFORMÁTICA, PUBLICACIONES y RECURSOS HUMANOS. Esta clasificación de los departamentos se representaría como una jerarquía

(también denominada generalización). Las generalizaciones nos proporcionan un mecanismo de abstracción que permite descomponer una entidad (que se denominará supertipo) en subtipos.

De esta forma vemos un conjunto de ejemplares de una entidad como de otra entidad. Así, por ejemplo, una "Persona" es un "Animal" y un "Reptil" es un "Animal"; en este caso, "Animal" puede considerarse el supertipo y "Persona" y "Reptil" son subtipos de "Animal". Los ejemplares o ejemplares de "Persona" lo son también de "Animal" e igual sucede con las de "Reptil".



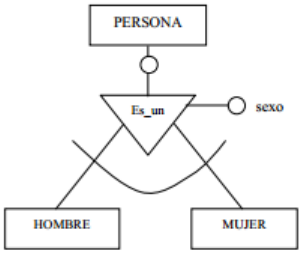
Tipos de interrelaciones jerárquicas o especialización:

- Una especialización **exclusiva**, denominada **sin solapamiento**, representa el hecho de que una instancia del supertipo sólo puede estar asociada a una y sólo una instancia de los subtipos.

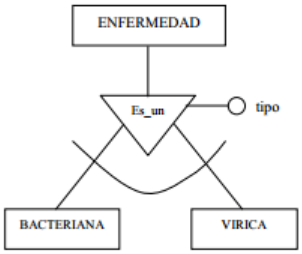
- Una especialización **inclusiva**, denominada **con solapamiento**, representa el hecho de que una instancia del supertipo puede estar asociada a la vez a varias instancias de cualquiera de los subtipos.

- Una especialización **total**, representa el hecho de que cualquier instancia del supertipo es de alguno de los subtipos de la especialización, no existiendo ocurrencias supertipos que no pertenezcan a alguno, varios o todos los subtipos.

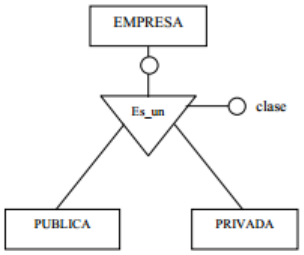
- Una especialización **parcial**, representa el hecho de que pueden existir ocurrencias del supertipo que no pertenezcan a ninguno de los subtipos de la especialización.



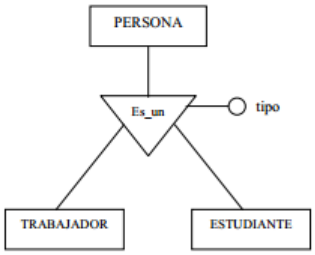
A) **Total sin solapamiento**: La entidad persona debe pertenecer a uno de los dos subtipos obligatoriamente (total) y además de forma exclusiva. Es decir, no podrá pertenecer a los dos a la vez (sin solapamiento)



B) **Parcial sin solapamiento**: La entidad enfermedad puede pertenecer a uno de los dos subtipos o a ninguno (parcial), ya que pueden existir enfermedades que no sean ni víricas ni bacterianas. Además de forma exclusiva. Es decir, no podrá pertenecer a los dos a la vez (sin solapamiento)



C) **Total con solapamiento**: La entidad empresa debe pertenecer a uno de los dos subtipos obligatoriamente (total) y no de forma exclusiva. Es decir, podrá pertenecer a los dos a la vez (con solapamiento). Para el caso de empresas públicas con gestión privada (semipúblicas o semiprivadas).



D) **Parcial con solapamiento**: La entidad persona no tiene que pertenecer a uno de los dos subtipos obligatoriamente (parcial) ya que una persona puede no trabajar ni estudiar. Pero también se puede dar el caso de que estudie y trabaje a la vez. Es decir, podrá pertenecer a los dos subtipos a la vez (con solapamiento)

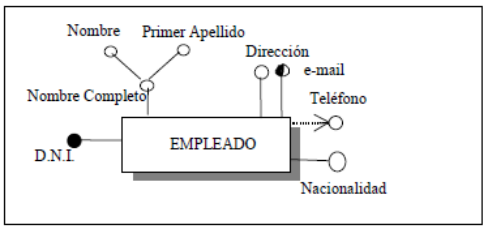
Fuente de las imágenes: http://www.ies-bezmiliana.org

Atributos

En este apartado ampliaremos nuestro conocimiento acerca de las restricciones semánticas sobre los atributos de las entidades y de las interrelaciones para de esta forma poder representar más fielmente los requisitos que nos piden para el diseño de una determinada base de datos.

Supongamos que en la entidad Empleado queremos recoger que un empleado puede tener más de un teléfono, tendríamos un atributo Teléfonos que tendría cero o más valores, esto es lo que llamamos atributo multivaluado.

Otro tipo de atributo es el atributo compuesto, que representa una agregación de atributos simples. Vamos a modificar el atributo Nombre de la entidad EMPLEADO ya que queremos un atributo, Nombre Completo, compuesto por Nombre y Primer Apellido.



2. Fase de Diseño Lógico Estándar: Modelo Relacional

2.1. Introducción

El modelo relacional fue propuesto en 1970 como alternativa a los modelos existentes hasta ese momento por el investigador Edgar Codd, tomando como base la teoría matemática de las relaciones. Este modelo perseguía la sencillez de comprensión y de manipulación de la base de datos por parte del usuario final.

En un principio el modelo era simplemente un modelo teórico objeto de investigación por parte de diversos centros, pero a medida que la tecnología fue evolucionando fueron surgiendo los primeros SGBD que lo soportaban y hoy en día el modelo relacional es el más conocido y difundido por los sistemas comerciales (ORACLE, INFORMIX,…).

La parte estática del modelo relacional consiste en la definición de los objetos permitidos. En la parte dinámica se definen las operaciones que se pueden realizar con estos objetos y sobre la base de datos.

Para que un SGBD pueda implementar un modelo de datos necesita apoyarse en un lenguaje de programación que gestione dicho modelo. El lenguaje aceptado por todos los sistemas de bases de datos comerciales es el SQL, lenguaje estructurado de consulta que permite crear, manipular y obtener datos de todos los objetos que constituyen la base de datos. Este lenguaje fue normalizado por la Organización Internacional de Estándares (ISO) en 1992 y sufre, como todos los estándares, revisiones periódicas que hacen que continuamente se editen anexos de la

norma. No obstante y como casi todas las normas, no está soportada por ningún producto comercial, es decir, cada SGBD incorpora su propio SQL que recoge, dependiendo de cada sistema, lo más significativo del estándar.

2.2. Elementos básicos del Modelo Relacional

Los elementos básicos del modelo relacional son las relaciones, cuya representación gráfica se realiza en forma de tabla. En las relaciones se pueden distinguir varios tipos de elementos: su nombre, los atributos que contiene y que representan las columnas de la tabla, y las tuplas o filas de la tabla. Además también podemos incluir los dominios que son aquellos conjuntos de donde los atributos toman los valores.

Supongamos que queremos modelar una base de datos para mantener información acerca de los museos de la Comunidad de Madrid y las exposiciones que en ellos se pueden visitar, a través del modelo de datos relacional . De cada museo se desea almacenar su nombre, la dirección, la zona (Madrid, El Escorial, etc) y el número de salas que contiene.

MUSEO

Nombre	Dirección	Zona	Salas
Arqueológico	Serrano, 13	Madrid	43
Centro de Arte Reina Sofía	Santa Isabel, 52	Madrid	21
Universidad de Alcalá de Henares	Plaza de San diego , s/n	Alcalá de Henares	7

El nombre de la relación de la figura es MUSEO, los atributos o columnas son "Nombre", "Dirección", "Zona" y "Sala". Las tuplas o filas son (Arqueológico; Serrano, 13; Madrid; 43), (Centro de Arte Reina Sofía; Santa Isabel, 52; Madrid; 21) y (Universidad de Alcalá de Henares; Plaza de San Diego, s/n; Alcalá de Henares; 7). El atributo "Zona" pertenece a un dominio Zonas = (Madrid, Alcalá de Henares, Torrejón de Ardoz,…).

Dentro de una relación podemos distinguir además dos nociones: el grado y la cardinalidad. El grado es el número de atributos (columnas) que posee la relación, en el caso de nuestro ejemplo el grado sería cuatro. La cardinalidad se refiere al número de tuplas (filas) de la relación, en nuestro caso tres.

Hasta ahora hemos hablado de los elementos básicos del modelo, hemos dicho que las relaciones son tablas, pero estas tablas tienen sus limitaciones, no son las tablas que normalmente manejamos; por ejemplo, en estas tablas se prohíbe que exista más de un valor en cada celda, este tipo de prohibiciones pertenecen a las restricciones inherentes al modelo y las comentaremos más adelante en la siguiente sección.

2.3. Estática del Modelo Relacional

Una relación, como se dijo en el apartado anterior se compone de un nombre, unos atributos (con sus correspondientes dominios) y un conjunto de tuplas. Es necesario distinguir entre lo que es la relación en sí y su contenido, es decir, una relación se define mediante un nombre y unos atributos de la siguiente manera:

NOMBRE_RELACIÓN (Atributo1: dominio1, Atributo2: dominio2,……, AtributoN: dominioN)

Los dominios son opcionales, pues los atributos pueden tomar valores en un conjunto acotado definido previamente, o bien y en principio, ser infinitas las posibilidades de un valor determinado.
El cuerpo de una relación, es decir, su contenido, es el conjunto de tuplas de la relación que, por su propia naturaleza varía con el tiempo, pues no tendría mucho sentido disponer de una base de datos que almacenara en todo momento la misma información, que esta información no variara con el tiempo, ya que en este caso con disponer de ficheros aislados para almacenarla podríamos tener suficiente.

Definición de relación:
ARTISTA (Nombre; Nacionalidad: Nacionalidades; Especialidad: Especialidades)

Contenido de la relación:

ARTISTA

Nombre	Nacionalidad	Especialidad
Agustín Redondella	Nacional	Pintura
Dolores Dahlahaus	Internacional	Fotografía
Mario Scifano	Internacional	Fotografía

Hasta este momento únicamente hemos hablado de los objetos del modelo relacional, pero no hemos hecho mención de cómo interactúan estos objetos en la base de datos, ni de cómo diseñar realmente una base de datos en dicho modelo. Para ello debemos previamente comentar las restricciones, tanto las inherentes al modelo, como las restricciones de usuario, que son aquellas que permiten recoger con la mayor fidelidad posible el mundo real objeto de nuestro diseño.

Restricciones inherentes.-

1. Ningún atributo puede tomar más de un valor para cada tupla.

   Esto es lo mismo que decir que en cada una de las celdas de una tabla que represente una relación no puede haber más de un valor. Así por ejemplo, no sería válido que en la relación ARTISTA el artista Mario Scifano tuviera como especialidades Fotografía y Escultura. Otra cosa distinta sería que en nuestro mundo real un artista pudiera tener varias especialidades, en cuyo caso deberíamos modelar la relación de otra manera, por ejemplo

creando otra donde se almacenaran todas las posibles especialidades y relacionándola con la primera, como veremos más adelante en el capítulo.

2. No importa el orden ni de las tuplas ni de los atributos.

3. Todas las tuplas de una relación deben ser distintas.

   Pues bien, qué ocurriría si en nuestro mundo real existiesen dos artistas con el mismo nombre, nacionalidad y especialidad. No piense el lector que este supuesto es tan descabellado, por ejemplo podría darse el caso de un artista contemporáneo que coincidiera en nombre con otro clásico (al no almacenarse la fecha de nacimiento no habría forma de distinguirlos). Podríamos actuar de forma negligente y no almacenar uno de los dos, con lo cual la obra de este autor quedaría sin reflejarse. Otra posibilidad sería pensar en añadir un nuevo atributo que identificara a cada artista unívocamente dando valores distintos a cada tupla, como puede ser un "Código de Artista". De esta forma cada tupla sería distinta de las demás.

   Este atributo que identifica unívocamente cada tupla de una relación es lo que se denomina en el modelo relacional **clave primaria** y en SQL se define como "PRIMARY KEY". Para representar la clave primaria de una relación, se suele poner en negrita el o los atributos implicados en dicha clave. Algún lector avispado podría discrepar de la opción escogida para identificar a los artistas de nuestra relación, pues para qué se elige introducir un atributo cuyo contenido es tan pobre semánticamente hablando si, habiendo escogido la "Fecha de nacimiento", éste, junto con el atributo "Nombre" identificarían por completo todas las tuplas. La justificación de nuestra elección radica en motivos de eficiencia, es mucho más eficiente a la hora de almacenar nuestra base de datos, una clave primaria corta en el sentido de que va a ser numérica o alfanumérica de pocos caracteres, mientras que el nombre y la fecha ocuparían más. Hemos justificado la razón de nuestra elección de clave primaria, no obstante "Nombre" y
   "Fecha de nacimiento" sería también un identificador válido para la relación. Este identificador se denomina en el modelo relacional **clave alternativa**. Es decir, una clave alternativa es aquella que aun pudiendo haber sido escogida como clave primaria de una relación, no lo ha sido por otros motivos, como pueden ser motivos de eficiencia, etc.

   Es importante aclarar que se ha hablado de clave primaria y clave alternativa y en el ejemplo hemos mencionado a dos atributos, esto es porque la clave puede estar formada por varios atributos pero siempre será una sola clave. Una clave primaria es aquella que identifica cada tupla de una relación de manera mínima, es decir, podríamos encontrarnos por ejemplo, cuatro atributos de una relación que identifiquen perfectamente a cada tupla, pero si con solo tres de ellos también las podemos identificar, la clave estará formada por esos tres.

4. Regla de integridad de la entidad.

   Esta regla hace referencia a que ningún atributo que forme parte de la clave primaria puede tomar un valor nulo.

Restricciones de usuario.-

1. Valores de uno o varios atributos que no pueden repetirse.

   Supongamos que, aun con todas las aclaraciones hechas en el apartado de restricciones inherentes, el usuario decide que no puede haber dos artistas con el mismo nombre, esto haría que el atributo "Nombre" fuera único para cada tupla.
   De aquí se deduce que toda clave primaria es única también, pero el lenguaje SQL mediante su definición de clave primaria ya lo contempla.

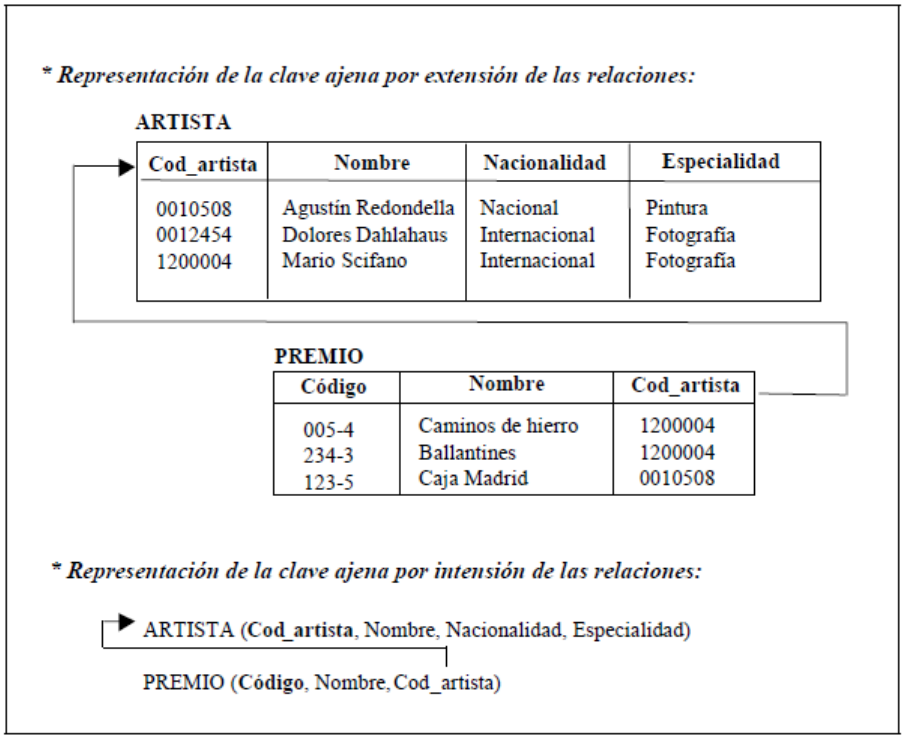
2. Atributos que deben tener siempre valores para todas las tuplas de la relación.

   Por ejemplo, y en el caso que nos ocupa, supongamos que no está permitido que un artista no tenga especialidad definida, esto significaría obligar a que el atributo especialidad tomara siempre valores, y el SQL lo reflejaría a través de "NOT NULL".

   El hecho de encontrarnos con atributos que no tengan valores es muy frecuente.
   Supongamos que en la relación ARTISTA queremos almacenar también la extensión de la obra de cada autor. Es claro que en muchos casos este dato no se conocerá, por tanto tendríamos entradas en la tabla cuya celda correspondiente a la obra estaría vacía.
   La obligatoriedad de valores en un atributo es también una propiedad de la clave primaria, por su propia definición.

3. Integridad Referencial.

   Se podría decir que la integridad referencial trata la forma en la que los datos de dos o más tablas se deben relacionar para no atentar contra la integridad de la base de datos, es decir, para evitar que haya información no necesaria.
   La integridad referencial se controla mediante lo que se denomina **clave ajena**. Una clave ajena de una tabla es un atributo o un conjunto de atributos, que es clave candidata18 de otra tabla con la cual está relacionada la primera, en SQL se define como "FOREIGN KEY".
   Para aclarar conceptos supongamos que se desean almacenar en la base de datos de nuestro ejemplo los premios que les han sido concedidos a los artistas por la Real Academia y, teniendo en cuenta que un premio es otorgado a un único artista. Esta relación PREMIO aparece un la siguiente figura. Se puede observar que posee un atributo "Artista" que almacena el artista al que se le ha concedido cada premio, y que coincide con la clave primaria de la relación ARTISTA ("Cod_artista"); "Artista" es clave ajena de la relación PREMIO. Se representa mediante una flecha que sale de la clave ajena y apunta hacia la clave primaria de la tabla con la que se relaciona.



**** Representación de la clave ajena por extensión de las relaciones:***

ARTISTA

Cod_artista	Nombre	Nacionalidad	Especialidad
0010508	Agustín Redondella	Nacional	Pintura
0012454	Dolores Dahlahaus	Internacional	Fotografía
1200004	Mario Scifano	Internacional	Fotografía

PREMIO

Código	Nombre	Cod_artista
005-4	Caminos de hierro	1200004
234-3	Ballantines	1200004
123-5	Caja Madrid	0010508

**** Representación de la clave ajena por intensión de las relaciones:***

ARTISTA (**Cod_artista**, Nombre, Nacionalidad, Especialidad)

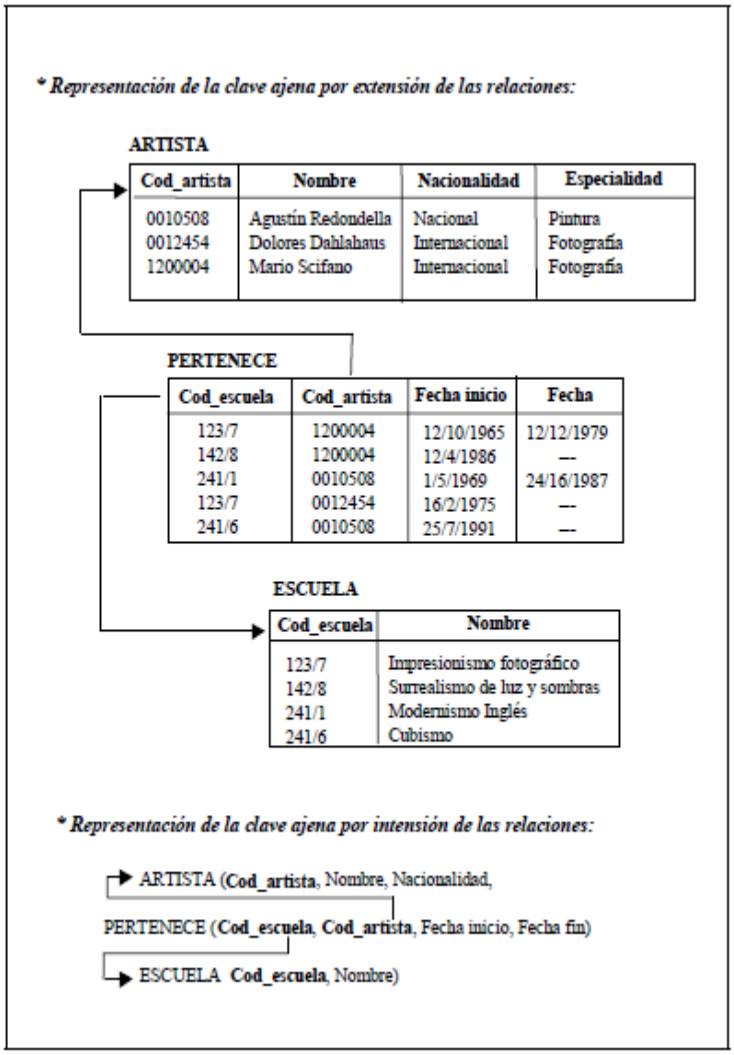
PREMIO (**Código**, Nombre, Cod_artista)

Existen situaciones en los que el control de la integridad referencial no es tan sencillo como en el caso anterior. Supongamos que se desean almacenar las Escuelas a las que pertenece cada artista y la relación que existe entre cada escuela y los artistas de nuestra base de datos, sabiendo que a una misma escuela pueden pertenecer varios artistas y que un artista puede haber pertenecido a escuelas diferentes (en distintos periodos de tiempo); en primer lugar debemos crear una relación ESCUELA cuyos atributos serán "Cod_escuela" (que será la clave primaria de la relación) y "Nombre". Para relacionar los artistas con las escuelas, en principio podemos pensar en varias soluciones:

Podríamos introducir el atributo Cod_artista en la relación ESCUELA como clave ajena que llamara a la relación ARTISTA, pero pensemos en las filas de la tabla resultante. Por cada escuela solo tendríamos un artista que perteneciera a ella, puesto que el modelo relacional no admite más de un valor en cada celda, lo que no es correcto. Podríamos pensar en el caso simétrico e introducir el atributo Cod_escuela en la relación ARTISTA. Esto significaría que cada artista solo puede pertenecer a una escuela, supuesto que va en contra de lo que se pide en el enunciado.

¿Qué hacer entonces?

El problema se resuelve introduciendo una nueva relación PERTENECE cuyos atributos sean "Cod_artista" y "Cod_escuela". Además estos atributos formarán en conjunto la clave primaria y de esta forma tendremos a cada artista con las distintas escuelas a las que pertenece y cada escuela con todos sus artistas. Podemos incluir dos nuevos atributos "Fecha inicio" y "Fecha fin" 19, para recoger el periodo de tiempo en que cada artista pertenece a una escuela determinada. Por último, tanto "Cod_artista" como "Cod_escuela" por separado, son claves ajenas de las relaciones ARTISTA y ESCUELA respectivamente.



** Representación de la clave ajena por extensión de las relaciones:*

ARTISTA

Cod_artista	Nombre	Nacionalidad	Especialidad
0010508	Agustín Redondella	Nacional	Pintura
0012454	Dolores Dahlahaus	Internacional	Fotografía
1200004	Mario Scifano	Internacional	Fotografía

PERTENECE

Cod_escuela	Cod_artista	Fecha inicio	Fecha
123/7	1200004	12/10/1965	12/12/1979
142/8	1200004	12/4/1986	---
241/1	0010508	1/5/1969	24/16/1987
123/7	0012454	16/2/1975	---
241/6	0010508	25/7/1991	---

ESCUELA

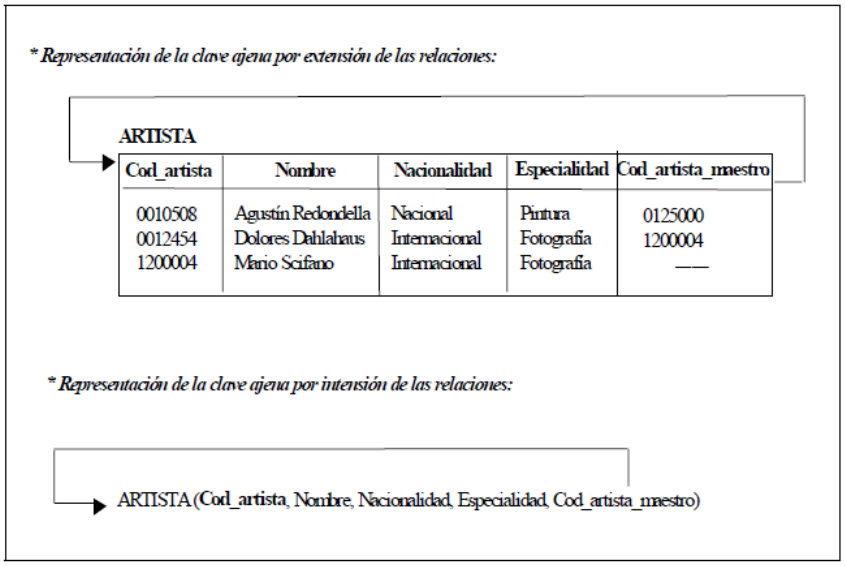
Cod_escuela	Nombre
123/7	Impresionismo fotográfico
142/8	Surrealismo de luz y sombras
241/1	Modernismo Inglés
241/6	Cubismo

** Representación de la clave ajena por intensión de las relaciones:*

ARTISTA (**Cod_artista**, Nombre, Nacionalidad,
PERTENECE (**Cod_escuela**, **Cod_artista**, Fecha inicio, Fecha fin)
ESCUELA **Cod_escuela**, Nombre)

Hasta el momento hemos visto como se pueden interrelacionar dos relaciones distintas mediante la definición de clave ajena, pero qué sucede en el caso de que una relación deba interrelacionarse con ella misma, esta situación, aunque en principio pueda parecer atípica, ocurre muchas veces en el mundo real. Supongamos por ejemplo, que en el caso de nuestros artistas, se desea recoger la relación que existe entre un artista y su maestro, en el caso de que exista; un maestro a su vez es un artista y por tanto tiene todas las cualidades (atributos) de ellos.

Podríamos introducir un atributo "Cod_artista_maestro" en la relación ARTISTA de manera que fuera clave ajena de ARTISTA que referencia a la misma relación ARTISTA.



Como consecuencia de la introducción de la noción de clave ajena en las relaciones, se pueden definir ciertas operaciones de borrado y modificación que permiten mantener la integridad de los datos de nuestras bases de datos. Estas operaciones son las siguientes:

A) Operación restringida ("**NO ACTION**" o "RESTRICT").

El borrado (DR) o la modificación (UR) de las filas de la relación que contiene la clave ajena no se permite mientras existan tuplas en la relación a la que se referencia. Esta situación ocurre por ejemplo en la relación ARTISTA con la clave ajena "Cod_artista_maestro". Solamente eliminamos un artista cuando no tenga ningún maestro, en caso contrario se rechaza la operación.

También se podría dar este tipo de operación en el caso por ejemplo de que en nuestra base de datos de artistas se desee mantener siempre el histórico de todos los premios, en este caso no debemos permitir borrar ningún artista de la relación ARTISTA al que se le haya concedido algún premio, solo podremos eliminar a los artistas que no tienen premios.

B) Operación en cascada ("**CASCADE**").

En este tipo de operación, cuando se elimina (DC) o modifica (UC) una tupla de la relación que es referenciada, los cambios se transmiten en cascada a las tuplas de la relación que contiene la clave ajena cuyos valores se han modificado. Así por ejemplo en el caso del apartado anterior, utilizando este tipo de operación, si borramos un artista se borran también todos los datos de su maestro, por lo que no es la mejor opción. En cambio, si utilizamos la operación "CASCADE" en la relación PERTENECE, cuando eliminemos un artista, se eliminará la relación que existe entre este y la escuela a la que pertenecía, pero no los datos propios de la escuela.
Estos dos tipos de operaciones sobre la clave ajena son los más habituales y los que implementan la mayoría de los SGBD comerciales. Existen otros dos tipos de operaciones (puesta a nulos "SET NULL" y valor por defecto "SET DEFAULT") que por su complejidad y falta de uso en los SGBD no se van a definir aquí, no obstante hay una amplia bibliografía

sobre el tema, la cual aparece al final del capítulo y que desde aquí animamos a que se consulte.

2.4. Dinámica del Modelo Relacional

La parte dinámica del Modelo Relacional, al igual que la dinámica de cualquier modelo de datos, define las operaciones que se pueden hacer con la base de datos. Estas operaciones pueden ser de varios tipos:

A) Operaciones de **recuperación** de datos de la base de datos (consultas).

B) Operaciones de **actualización** de datos de la base de datos. Estas operaciones son:
- Inserción de tuplas en la base de datos.
- Modificación de tuplas (de algunos de sus valores)
- Borrado de tuplas.

Estas operaciones se expresan mediante lenguajes de manipulación relacionales. Los lenguajes de manipulación pueden dividirse en lenguajes navegacionales, en los que se debe indicar el camino para llegar al dato, y lenguajes de especificación, en los cuales se recupera la información mediante la imposición de condiciones y no indicando el camino. El lenguaje de manipulación de datos del SQL (LMD) es un lenguaje de especificación, que consiste en un conjunto de sentencias que nos permiten manipular la base de datos.

3. Transformación del modelo E/R al modelo relacional.

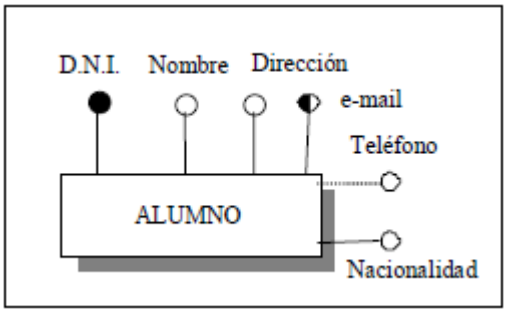
3.1. Introducción

La primera fase en el diseño de una base de datos es la de recoger las características que deseamos plasmar en la base de datos en un modelo conceptual, es decir, en un modelo sencillo, cercano a las personas no informáticas para que de esta forma puedan entender nuestro diseño y realizar la validación de este. Este modelo es el Entidad/Interrelación, que utiliza como elementos básicos las entidades (representadas por rectángulos), las interrelaciones (representadas por rombos) y los atributos (representadas con un círculo unido a una entidad o a una interrelación mediante una línea).

En la segunda fase del diseño tenemos que transformar el esquema realizado en el modelo Entidad/Interrelación a un esquema lógico, ya que no existe ningún Sistema Gestor de Bases de Datos que soporte el modelo E/R. Esta fase es muy importante ya que para pasar de un esquema a otro debemos preservar todas las características recogidas en la primera de fase. De esta forma nos aseguraremos que la Base de Datos que implementemos se corresponda con lo que en un principio queríamos recoger y almacenar.

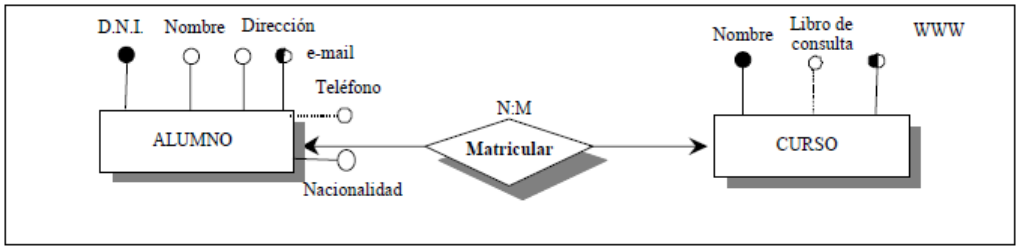
3.2. Reglas básicas para la transformación del modelo E/R al modelo relacional.

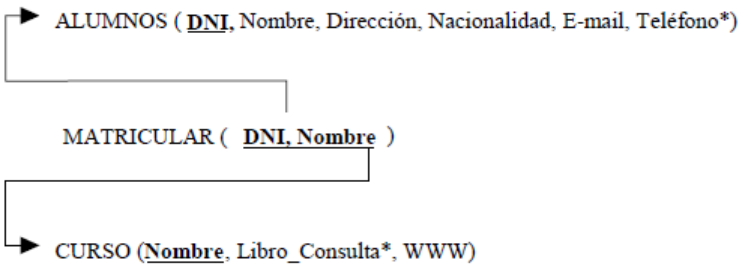
- La primera de las reglas de transformación nos dice que toda entidad se transforma en una relación o tabla, y los atributos o características asociadas a ella pasan a ser atributos de la relación.



ALUMNOS (**Dni**, E-mail, Nombre, Dirección, Nacionalidad, Teléfono*)

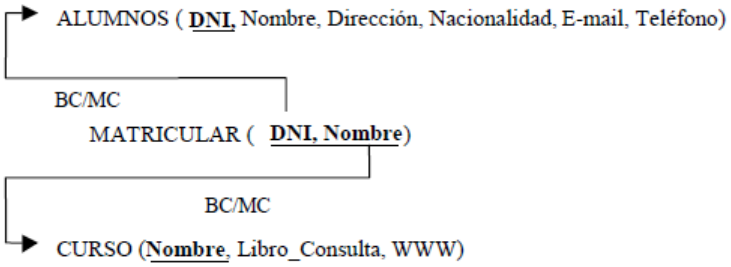
- La segunda regla de transformación nos indica que las interrelaciones cuyo tipo de correspondencia es N:M se transforman en una nueva relación cuyo nombre se corresponde con el nombre de la interrelación y donde la clave primaria se compone de los atributos identificadores de las dos entidades que relaciona.



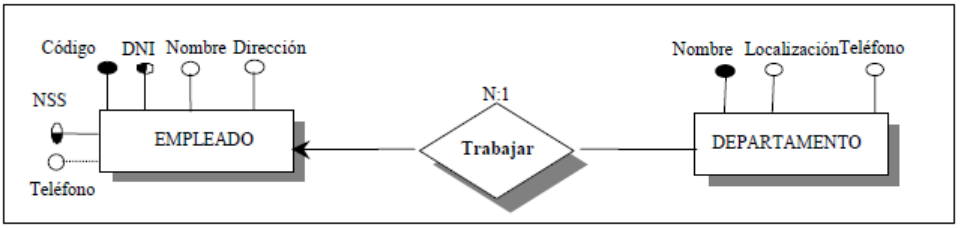


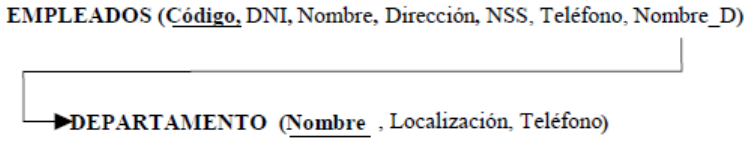
En el esquema relacional anterior nos faltaría representar las opciones de borrado y modificación que en este caso sólo pueden ser o restringidas o en cascada ya que, tanto el DNI como el Nombre al pertenecer a la clave primaria de la relación MATRICULAR no pueden tomar valores nulos ni valores por defecto.

Si borramos un alumno ¿borraríamos todos los cursos que ha realizado?. Para recoger esta característica deberíamos tomar la opción de borrado en cascada al igual que si quisiéramos que las modificaciones en la relación ALUMNO se propagaran a la relación MATRICULAR. Por el contrario, si no queremos borrar a los alumnos que tengan cursos asociados en MATRICULAR escogeríamos la opción de borrado restringida. Tomamos la primera alternativa explicada y razonamos de la misma forma para la clave ajena Nombre.



- La tercera y última regla nos indica que la transformación de interrelaciones cuyo tipo de correspondencia es 1:N se traduce en una propagación de clave o en una nueva relación si la interrelación posee atributos. Veamos en qué consiste el fenómeno de propagación de clave mediante el siguiente ejemplo.




EMPLEADOS (Código, DNI, Nombre, Dirección, NSS, Teléfono, Nombre_D)
DEPARTAMENTO (Nombre , Localización, Teléfono)