

Tema 2. Modelado Entidad-Relación

Durante la definición de requerimientos y el diseño conceptual hay que identificar las necesidades básicas en cuanto a datos, relaciones entre datos, así como las operaciones que se van a llevar a cabo sobre los datos. En este tema se estudia el concepto de modelo como técnica para la representación de la realidad, y se estudiará en profundidad el Modelo Entidad-Relación, definiendo su notación, introduciendo la definición de los distintos tipos de clave, y viendo como documentar los diagramas. Además, se describe el proceso de transformación a tablas de un diagrama entidad-relación de forma que podamos obtener un esquema para una base de datos relacional. También se estudiarán los conceptos de generalización, especialización y agregación, terminando con la presentación de los diagramas de clases UML.

2.1. Introducción al modelado conceptual de bases de datos

2.1.1. Realidad y modelos

El proceso de definición de requerimientos y del diseño conceptual exige la identificación de las exigencias de la información de los usuarios, así como representar los datos en un modelo bien definido. Esto se consigue estudiando la naturaleza de los datos y su significado en la base de datos.

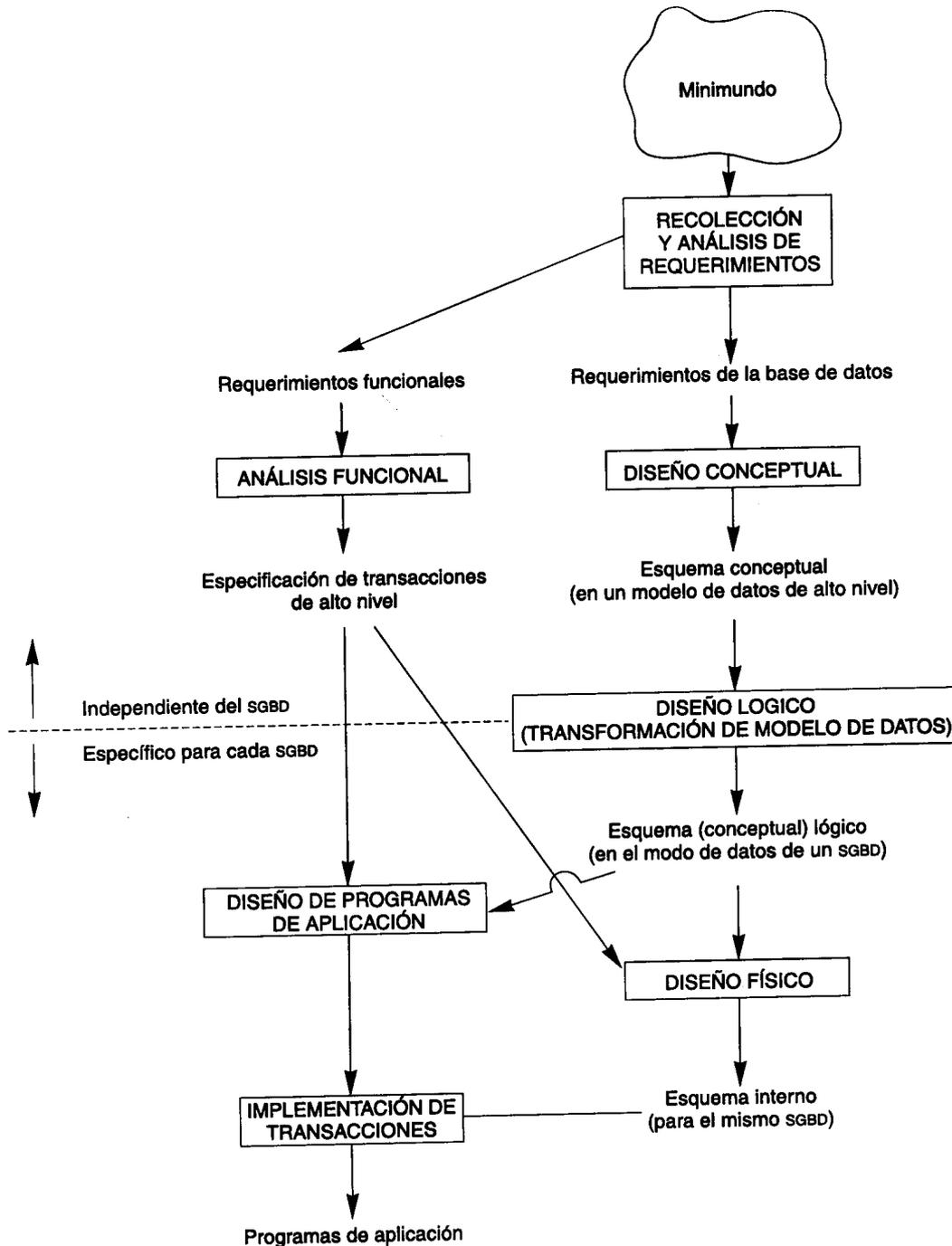
Un **modelo** es una representación de la realidad que sólo conserva los detalles relevantes. Una base de datos incorpora un modelo de la realidad.

El SGBD administra la base de datos de forma que cada usuario pueda acceder, manipular y modificar los datos que constituyen su modelo de la realidad. La utilización adecuada de estos datos puede conducir a la toma de una decisión muy importante, por lo que los modelos son herramientas muy potentes que eliminan los detalles irrelevantes y comprenden la realidad de los usuarios.

Por tanto, el proceso de modelado necesita de una *asociación* o *identificación* de elementos de la realidad con elementos del modelo. Si esta asociación se hace de forma correcta, se puede utilizar el modelo para resolver el problema, pero de lo contrario, el modelo puede producir una solución incorrecta.

2.1.2. Proceso de creación de una base de datos

La figura siguiente ilustra el proceso de diseño de una base de datos.



El primer paso que se realiza es la *recopilación de información y análisis de requerimientos*, durante el cual, los diseñadores entrevistan a los futuros usuarios del Sistema de Base de Datos para comprender y redactar los requerimientos de información. El resultado de este paso será un conjunto de requerimientos redactado de forma concisa.

Paralelamente a la especificación de requerimientos de datos, conviene especificar los *requerimientos funcionales* de la aplicación. Se trata de las operaciones de usuarios (transacciones) que se aplicarán a la base de datos e incluye la obtención y actualización de datos.

Una vez que se ha realizado el análisis de requerimientos, el paso siguiente es *crear el esquema conceptual* para la base de datos mediante un modelo de datos conceptual de alto nivel. A este paso se le denomina *diseño conceptual* de la base de datos. El esquema conceptual es una descripción concisa de los requerimientos de información de los usuarios, y contiene información sobre las relaciones y restricciones. Este esquema conceptual puede servir como punto de referencia para asegurarse de que se satisfacen todos los requerimientos de los usuarios.

Una vez que haya terminado el proceso de diseño hay que empezar a implementar el SBD con un SGBD comercial. La mayoría de los SGBD actuales utilizan un modelo de datos de implementación, por lo que hay que traducir el modelo de datos de alto nivel al modelo de datos de implementación, lo que se conoce como el proceso de *diseño lógico* de la base de datos o transformación de modelos de datos. El resultado es un esquema de la base de datos especificado en el modelo de datos de implementación del SGBD.

El último paso es la fase de *diseño físico* de la base de datos, durante la cual se especifican las estructuras de almacenamiento internas y la organización de los archivos de la base de datos. Además, se han de diseñar e implementar los programas de aplicación para las transacciones que se recopilaron en el análisis de requerimientos funcionales.

2.2. Conceptos básicos

El modelo E-R (Entidad-Relación) es un modelo de datos conceptual de alto nivel y que se suele utilizar bastante en el diseño de bases de datos. Se basa en una *percepción del mundo real* que consiste en un conjunto de objetos básicos denominados *entidades* y *relaciones*, y se desarrolló para facilitar el diseño de bases de datos.

El modelo E-R crea un modelo de la realidad que se corresponde con la parcela de la realidad que queremos modelar, y lo hace de forma que es independiente de la implementación posterior, ofreciendo un alto nivel de abstracción, y siendo una herramienta gráfica fácil de comprender.

El resultado del modelado E-R es un diagrama E-R que representa una estructura lógica general de la base de datos.

2.2.1. Entidades y conjuntos de entidades

Una *entidad* es un objeto que existe y que se puede distinguir de otros objetos, es decir, una "cosa" del mundo real con existencia independiente. (Por ejemplo, Juan López, con DNI 37465748 es una identidad, ya que identifica de forma única a una persona en la realidad que vayamos a modelar). Las entidades pueden ser concretas

(como una persona o un libro), o pueden ser abstractas (como un día festivo) o un concepto (un puesto de trabajo o un curso universitario).

Un *conjunto de entidades* es una agrupación de entidades del mismo tipo. (Por ejemplo, el conjunto de entidades CLIENTES estaría formado por todas las personas que tienen cuenta en un banco). Los conjuntos de entidades no tiene por que ser disjuntos, es decir, una entidad puede pertenecer a más de un conjunto de entidades (una persona que fuese empleado –pertenecería al conjunto de entidades EMPLEADOS- y que a su vez tuviese una cuenta en el banco –pertenecería al conjunto de entidades CLIENTES).

El concepto de conjunto de entidades se asemeja a la definición de tipos de los lenguajes de programación, mientras que el concepto de entidad se correspondería con las variables.

Por tanto, una base de datos incluye una serie de conjuntos de entidades, y cada uno de ellos puede contener un número arbitrario de entidades.

2.2.2 Atributos y Dominios

Una entidad está representada por un conjunto de *atributos*, por tanto, cada entidad tiene una serie de propiedades específicas (*atributos*) que la describen. (Por ejemplo, nombreCliente, seguridadSocial y ciudadCliente podrían ser atributos del conjunto de entidades EMPLEADOS). Para cada atributo hay un conjunto de valores permitidos, al que se denomina *dominio*. (Por ejemplo el dominio de nombreCliente sería el conjunto de todas las cadenas de texto con una longitud determinada).

A continuación se muestra parte de una base de datos para un ejemplo bancario.

JUAN PEREZ	27364756	PINO, 4. ALMERÍA
ANA CARO	28643785	ANCHA, 7. GRANADA
ANGEL MURO	53644753	INGLES, 3. JAEN
JOSE LOPEZ	26743845	ALTA, 9. HUELVA

234	200
523	45644
254	65777
286	100477
244	98675

CLIENTES

CUENTAS

2.2.3. Relaciones y conjuntos de relaciones

Una *relación* es una asociación entre varias entidades. (Por ejemplo, podemos definir una relación que asocia al cliente Juan Pérez la cuenta 254).

Un *conjunto de relaciones* es un grupo de relaciones del mismo tipo. Formalmente es una relación matemática con $n \geq 2$ conjuntos de entidades y puede ser obligatorio u opcional. ((Juan Pérez, 254) sería una relación).

Una *relación binaria* es una relación que asocia a dos conjuntos de entidades, siendo de este tipo la mayoría de los conjuntos de relaciones de un Sistema de Bases de Datos. Sin embargo, a veces hay conjuntos de relaciones que asocian más de dos conjuntos de entidades. A las relaciones entre tres conjuntos de entidades se les denominan *relaciones ternarias*, pero siempre es posible sustituir un conjunto de relaciones no binarias por varios conjuntos de relaciones binarias distintos (promoviendo el conjunto de relaciones a un conjunto de entidades).

2.2.4 Relaciones recursivas y papel de una entidad en una relación

Todos los conjuntos de entidades que participen en un conjunto de relaciones desempeñan un papel concreto en la relación. Por tanto, un *papel* es la función que juega una entidad en una relación. (Por ejemplo, el conjunto de relaciones TRABAJA podría modelarse mediante pares de entidades sucursal y empleado. Así pues, sucursal juega el papel de sucursal y empleado juega el papel de empleado). Cabe decir que el papel suele estar implícito, y que juega un papel especialmente importante para relaciones entre un mismo conjunto de entidades. A este tipo de relaciones se les denomina *relaciones recursivas*.

2.2.5 Atributos descriptivos

Una relación puede tener atributos similares a los de los conjuntos de entidades. A este tipo de atributos se les denomina *atributos descriptivos*. (Por ejemplo, si estuviésemos en un problema en que los empleados pudiesen trabajar en varias empresas, y en cada una de ellas podrían desempeñar un cargo diferente, el atributo correspondiente al cargo sería un atributo descriptivo).

2.3. Tipos de relaciones con grado mayor que dos

Supongamos que tenemos el siguiente ejemplo. Disponemos de una serie de proveedores que venden sus productos a diferentes clientes. Si estamos interesados en conocer qué producto es vendido a cada cliente por cada proveedor necesitaríamos una relación ternaria. Si modelásemos esto mediante relaciones binarias estaríamos expresando que los proveedores venden productos, y que los productos son vendidos a los proveedores. Sin embargo, no se estaría representando el hecho que expresa de qué proveedor es el producto vendido, sino que únicamente se estaría indicando que los clientes compran productos, independientemente del proveedor que los fabrique.

Este tipo de relaciones, ya sean de grado 3 o superior obligan a que participe al menos una entidad de cada uno de los conjuntos de entidades involucrados en la relación.

2.4. Restricciones de cardinalidad

Los conjuntos de relaciones suelen tener ciertas restricciones, como el cardinal de asignación, que limita el número entidades de un conjunto de entidades con las que se puede asociar una entidad de otro conjunto de entidades.

Para un conjunto de relaciones binario entre dos conjuntos de entidades A y B , el cardinal de asignación tiene que ser una de las siguientes:

- **Uno a uno (1:1).** Una entidad de A está relacionada con a lo sumo una entidad de B , y una entidad de B está asociada con a lo sumo una entidad de A .
- **Uno a muchos (1:M).** Una entidad de A está asociada con cualquier número de entidades de B , pero cada una de las entidades de B , sólo puede estar relacionada con a lo sumo una entidad de A .
- **Muchos a muchos (M:N).** Una entidad de A está relacionada con un número cualquiera de entidades de B , y una entidad de B puede estar relacionado con un número cualquiera de entidades de A .

En un ejemplo bancario podríamos decir que una persona puede tener varias cuentas y que una misma cuenta puede ser de varias personas; por tanto la relación que hay entre los conjuntos de entidades CUENTAS y CLIENTES es de muchos a muchos. En cambio, si consideramos a los empleados de una sucursal, observamos que en una sucursal trabajan varios empleados, pero que un empleado sólo puede trabajar en una sucursal; por tanto entre los conjuntos de entidades SUCURSALES y EMPLEADOS hay una relación uno a muchos.

2.5 Restricciones de integridad

Para completar los aspectos del mundo real representados en un modelo, necesitamos poder expresar *restricciones* de los datos que van más allá de las limitaciones impuestas por las definiciones de entidades, atributos y relaciones. Estas restricciones pueden ser:

- Restricciones de clave. Las claves son atributos o conjuntos de atributos que identifican una entidad dentro de su conjunto.
- Restricciones de valor único. Exigen que a un valor de un atributo o conjunto de atributos se le asocie un valor único de otros atributos.
- Restricciones de integridad referencial. Exigen que un valor referenciado por alguna entidad exista realmente en la base de datos.
- Restricciones de dominio. Exigen que el valor de un atributo pertenezca a un conjunto específico de valores.
- Restricciones generales. Son condiciones arbitrarias que deben cumplirse en la base de datos.

2.6 Claves. Tipos de entidades

Los conjuntos de entidades suelen tener un atributo cuyo valor es único para cada una de las entidades del conjunto. A este tipo de atributos se les denomina atributos clave e identifican de forma única a cada entidad.

2.6.1. Tipos de claves

- **Superclave.** Conjunto de atributos que considerados conjuntamente permiten identificar de forma única a una entidad en el conjunto de entidades (por ejemplo, un atributo dniEmp permite distinguir a una entidad Empleado de otra. Asimismo, el conjunto de atributos {dniEmp, nombreEmp} también permitiría distinguir a una entidad Empleado de otra. Por tanto, ambos ejemplos serían dos ejemplos de superclaves). El problema es que las superclaves pueden tener atributos innecesarios, es decir, si K es una superclave, también lo es cualquier superconjunto de atributos que incluya a K .
- **Clave candidata.** Una superclave mínima, es decir, una superclave a la que si le quita un atributo deja de ser superclave (por ejemplo, el atributo dniEmp y el conjunto de atributos nombreEmp y teléfono podrían ser dos ejemplos de claves candidatas).
- **Clave primaria.** Clave que elige el diseñador de la base de datos para la identificación de entidades dentro de un conjunto de entidades.

2.6.2. Tipos de entidades

Es posible que un conjunto de entidades no tenga un conjunto de atributos suficiente para crear una clave primaria, por lo que clasificaremos los conjuntos de entidades dependiendo de si tienen o no claves primarias.

- **Conjunto de entidades débil.** Conjunto de entidades que no tiene clave primaria.
- **Conjunto de entidades fuerte.** Conjunto de entidades que tiene clave primaria.

Por ejemplo, en el caso de un banco, podríamos hablar del conjunto de entidades Transacciones como un conjunto de entidades con tres atributos: numeroTrans, Fecha e Importe. Aunque cada entidad transacción es diferente, las transacciones de cuentas diferentes pueden tener el mismo número de transacción. Por tanto, este conjunto de entidades no tiene clave primaria, y por tanto es débil.

Un conjunto de entidades débil sólo debe formar parte de una relación 1:M y no debe tener atributos descriptivos.

2.6.3. Dependencia por existencia

X depende por existencia de Y , si la existencia de X está supeditada a la existencia de Y . Esto quiere decir que si se elimina Y , también se eliminará X . A la entidad Y se le denomina *entidad dominante* y a la entidad X se le denomina *entidad subordinada*. En el ejemplo anterior, el conjunto de entidades TRANSACCIONES depende por existencia de CUENTAS, ya que si se elimina una entidad cuenta, también se eliminarán todas las transacciones relacionadas con dicha cuenta.

Los conceptos de conjuntos de entidades fuerte y débil están relacionados con el concepto de dependencia por existencia, y aunque no existe una forma de diferenciar a las entidades débiles (no tienen clave primaria) tenemos que disponer de un medio para poder identificar a una entidad por su relación con la entidad de la que depende.

Al conjunto de atributos que permite identificar a las entidades débiles relacionadas con una entidad fuerte se le denomina *discriminador* (por ejemplo, el discriminador del conjunto de entidades débil Transacciones sería numeroTrans). La clave primaria de un conjunto de entidades débil está formada por la clave primaria del conjunto de entidades del que depende y por su discriminador (por ejemplo, la clave primaria del conjunto de entidades débil Transacciones sería {numeroCta, numeroTrans}).

2.6.4. Atributos y claves de un conjunto de relaciones

Al igual que tenemos un mecanismo para distinguir entidades dentro de un conjunto de entidades, tenemos que tener otro mecanismo para poder diferenciar a cada una de las relaciones de un conjunto de relaciones.

El conjunto de atributos de una relación está formado por la concatenación de los atributos de las claves primarias de los conjuntos de entidades que participan en la relación, junto con los atributos descriptivos de la relación.

La determinación de la clave de un conjunto de relaciones depende del tipo de relación que exista entre los conjuntos de entidades.

Si el conjunto de relaciones no tiene atributos descriptivos, entonces el conjunto de atributos forma una superclave. Esta superclave es clave primaria si la relación es muchos a muchos, pero si la relación es muchos a uno, la clave primaria es la clave del conjunto de entidades de la parte muchos.

Por ejemplo, en el caso de empleados que trabajan en empresas y su cargo varía en función de la empresa, el conjunto de relaciones TRABAJA estaría formado por los atributos dniEmp, cifEmpresa y cargo. En el caso de una relación muchos a muchos, la clave primaria sería dniEmp y cifEmpresa. Sin embargo, si la relación fuese uno a muchos, la clave primaria estaría formada sólo por dniEmp, sin que hiciese falta el atributo cifEmpresa, ya que una persona sólo trabajaría en una empresa.

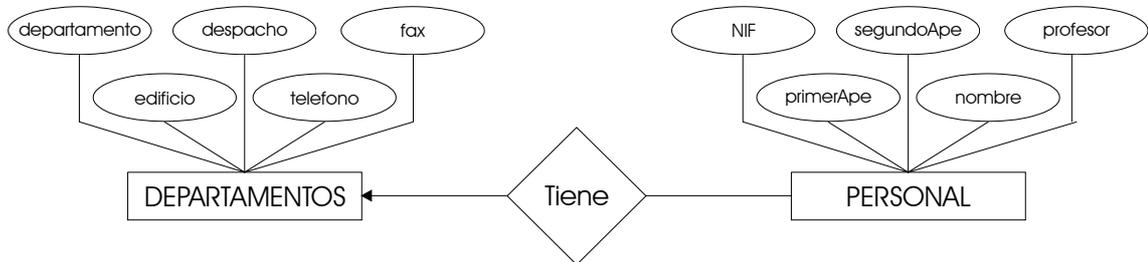
2.7. Diagramas Entidad-Relación. Notaciones

Como vimos en el tema de introducción, la estructura lógica de una base de datos se puede representar mediante un diagrama Entidad-Relación, que se construye a partir de los componentes siguientes:

- Rectángulos: Representan conjuntos de entidades.
- Elipses: Representan atributos.
- Rombos: Representan conjuntos de relaciones.

- Líneas: Se utilizan para unir los tres componentes anteriores.

El diagrama E-R representa las restricciones de cardinalidad con puntas de flecha dirigidas a los conjuntos de entidades que participan como uno en una relación (por ejemplo, un departamento está compuesto por personal docente e investigador que sólo pueden pertenecer a un departamento).



En este tema trabajaremos con una base de datos de ejemplo, llamada BANCO, que servirá para ilustrar los conceptos del modelo E-R y su uso en el diseño de esquemas. En primer lugar mencionaremos los requerimientos de información de esta base de datos, y a continuación crearemos su esquema conceptual paso a paso, al tiempo que se presentan los conceptos de modelado del modelo E-R. La base de datos BANCO se ocupa de las sucursales, clientes, empleados, cuentas y operaciones que se realizan en el banco.

Supongamos que una vez terminada la fase de recopilación y análisis de requerimientos, los diseñadores de la base de datos crearon la siguiente descripción:

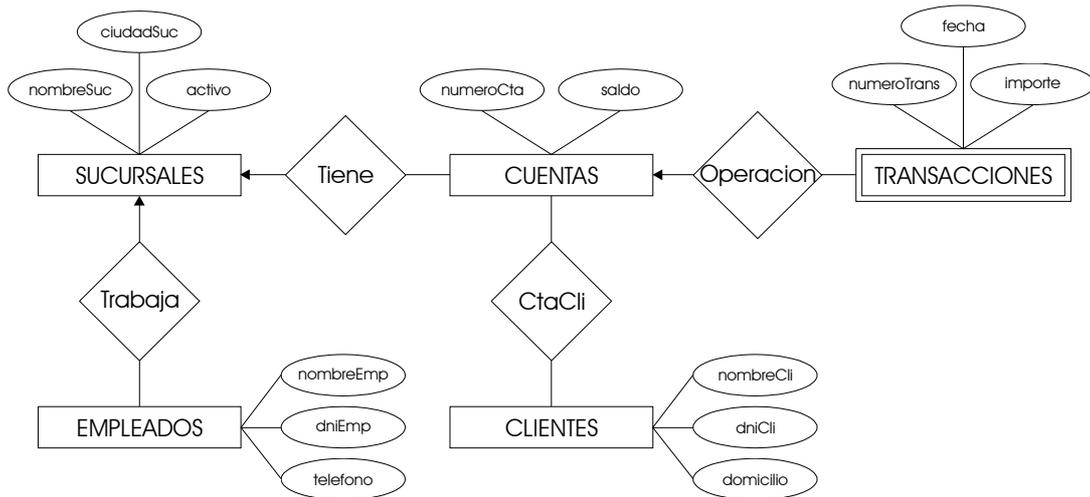
El banco está organizado en sucursales, cada una de las cuales tiene un nombre único, está situada en una ciudad y tiene un activo determinado. Además, los empleados de una sucursal tienen un nombre un DNI y un teléfono de contacto. Nos interesa saber en qué sucursal trabaja un empleado

Cada sucursal tiene un número indeterminado de cuentas, donde cada una de ellas tiene un número de cuenta y un saldo. Nos interesa saber cuáles son las cuentas de una sucursal.

Cada cuenta corriente puede pertenecer a varios clientes del banco, y éstos tienen un nombre, un DNI y un domicilio. Además un cliente, como es natural puede tener varias cuentas en el mismo banco. Nos interesa saber cuáles son los titulares de una cuenta.

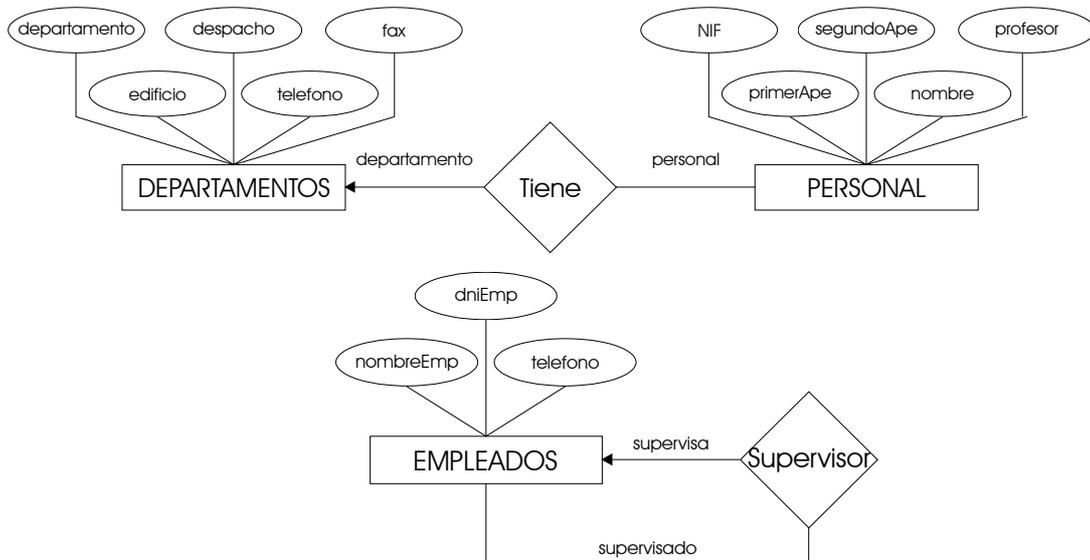
También se desean guardar las operaciones que se realizan sobre una cuenta, con un número de transacción, una fecha y el importe de la transacción

A continuación se muestra el diagrama ER para este ejemplo de bases de datos.



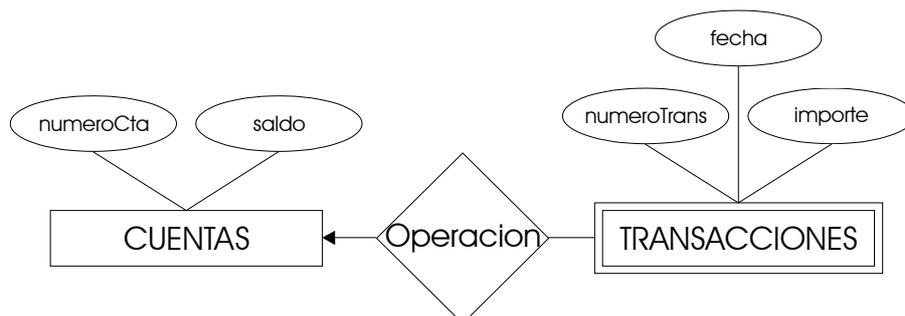
2.7.1. Notación para el papel de las entidades

En los diagramas E-R, los papeles que juega cada entidad en una relación se realiza escribiendo el papel entre la línea que une los rectángulos con los rombos.



2.7.2. Notación para las entidades débiles

Las entidades débiles se representan mediante un rectángulo con el contorno doble. A continuación se muestra el conjunto de entidades débil TRANSACCIONES que depende del conjunto de entidades fuerte CUENTAS a través del conjunto de relaciones OPERACION.



2.8. Documentación de diagramas Entidad-Relación

El modelo Entidad-Relación es una herramienta potente y expresiva, pero es prácticamente imposible que permita representar con detalle todos los aspectos de una aplicación, por diversas razones.

En primer lugar, en un diagrama E-R sólo aparecen los nombres de los diversos conceptos, y no su significado. Además, cuando un esquema es complejo, puede que no sea posible representar todas las propiedades de los elementos de forma comprensible, por cuestiones de legibilidad.

En segundo lugar, tenemos cuestiones de potencia expresiva. Hay casos en los que resulta imposible representar ciertas propiedades de los datos usando las herramientas ofrecidas por el modelo E-R, como la de que un profesor sólo puede ser director del departamento al que pertenece, o un empleado no puede cobrar más que el gerente. Estas propiedades se corresponden con restricciones de integridad, y se formulan mediante *reglas de funcionamiento de la organización*.

Por todo lo anterior, resulta indispensable completar un diagrama E-R con documentación complementaria, que permita facilitar la interpretación del esquema y describir propiedades de los datos que no pueden expresarse directamente con los elementos del modelo. Esta documentación debe incluir una descripción de los componentes del diagrama y las reglas de funcionamiento de la organización, y debe estar estructurada en un *diccionario de datos*.

2.9. Paso a tablas de los diagramas Entidad-Relación

Una base de datos que se represente mediante un diagrama E-R puede representarse mediante un conjunto de tablas. De forma general suele existir una tabla para cada conjunto de entidades y una tabla para cada conjunto de relaciones, asignándole a cada tabla el nombre del conjunto de entidades o del conjunto de relaciones correspondiente, aunque esto es sólo de forma general, por lo que habrá que ver las particularidades de cada caso. Cada tabla consta de una serie de columnas con un nombre único.

2.9.1 Representación de conjuntos de entidades fuertes

Sea F un conjunto de entidades fuertes con los atributos f_1, f_2, \dots, f_n . Representaremos a este conjunto de entidades mediante una tabla denominada F con n columnas que se corresponden con los n atributos de F , y donde cada fila de F se corresponde con una entidad de F .

Por ejemplo, la tabla que se corresponde con el conjunto de entidades *CLIENTES* sería una tabla de tres columnas: *nombreCli*, *dniCli* y *domicilio* y cada una de sus filas se corresponderían con las entidades cliente de la base de datos bancaria.

<i>nombreCli</i>	<i>dniCli</i>	<i>domicilio</i>
<i>Aitor</i>	<i>12345678</i>	<i>Palma 7</i>

<i>Nestor</i>	<i>23456789</i>	<i>Austria 12</i>
<i>Victor</i>	<i>34567890</i>	<i>Pipa 5</i>

2.9.2 Representación de conjuntos de entidades débiles

Sea D un conjunto de entidades débiles con los atributos d_1, d_2, \dots, d_n . Sea E el conjunto de entidades fuerte del que depende D . La clave primaria del conjunto de entidades fuerte consta de los atributos f_1, f_2, \dots, f_m . Representaremos el conjunto de entidades débil mediante una tabla denominada D con $m+n$ columnas que se corresponden con los atributos del conjunto:

$$\{f_1, f_2, \dots, f_m\} \cup \{d_1, d_2, \dots, d_n\}$$

Por ejemplo, la tabla que se corresponde con el conjunto de entidades *TRANSACCIONES* sería una tabla de cuatro columnas: *numeroCta* (la clave primaria del conjunto de entidades fuerte), *numeroTrans*, *fecha*, *importe* (los atributos del conjunto de entidades débil), y cada una de sus filas se corresponderían con las transacciones de las cuentas del banco.

<i>numeroCta</i>	<i>numeroTrans</i>	<i>fecha</i>	<i>Importe</i>
<i>1</i>	<i>1</i>	<i>16-10-97</i>	<i>+100</i>
<i>2</i>	<i>1</i>	<i>16-10-97</i>	<i>+200</i>
<i>1</i>	<i>2</i>	<i>17-10-97</i>	<i>-5000</i>
<i>3</i>	<i>1</i>	<i>20-10-97</i>	<i>+1000</i>

2.9.3 Representación de conjuntos de relaciones

Sea R un conjunto de relaciones que relaciona los conjuntos de entidades E_1, E_2, \dots, E_n , donde las claves primarias de los conjuntos de entidades vienen dados de esta forma:

Entidad	Clave primaria
E_1	$C1 = \{ a_1, a_2, \dots, a_{m1} \}$
E_2	$C1 = \{ a_1, a_2, \dots, a_{m2} \}$
...	...
E_n	$C1 = \{ a_1, a_2, \dots, a_{mn} \}$

Representaremos al conjunto de relaciones por una tabla con una serie de columnas que se corresponde con la unión de los atributos de las claves primarias que participan en la relación.

Por ejemplo, el conjunto de relaciones *CtaCli* vendría dado por una tabla con dos columnas: *numeroCta* (la clave primaria del conjunto de entidades *CUENTAS*) y *dniCli* (la clave primaria del conjunto de entidades *CLIENTES*) y cada una de las filas representaría a las cuentas que tiene cada cliente.

<i>numeroCta</i>	<i>dniCli</i>
1	12345678
2	12345678
2	23456789
3	34567890

2.9.4. Consideraciones sobre los conjuntos de entidades débiles

En el caso de los conjuntos de relaciones en los que participan conjuntos de entidades débiles, no se crea una tabla para el conjunto de relaciones porque los datos que guardaría el conjunto de relaciones ya están en el conjunto de entidades débil, y por tanto habría datos redundantes.

TRANSACCIONES: (numeroCta, numeroTrans, fecha, importe)

OPERACION: (numeroCta, numeroTrans)

Por tanto la tabla del conjunto de relaciones no se representa, y en su lugar se utiliza la del conjunto de entidades débiles.

2.9.5. Consideraciones sobre las relaciones uno a muchos

Algo parecido a las consideraciones sobre los conjuntos de entidades débiles, le ocurre a las relaciones uno a muchos.

Lo que se hace es que se crea una tabla para el conjunto de entidades que participa como uno y otra tabla para el conjunto de entidades que participa como muchos, incluyendo esta última todos sus atributos y la clave primaria del conjunto de entidades que participa como uno.

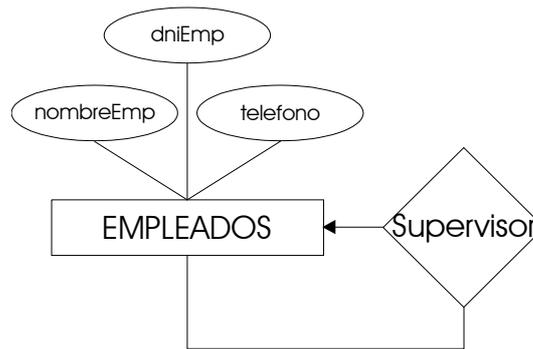
Por ejemplo, en el caso de los departamentos y profesores los esquemas de las tablas serían los siguientes:

DEPARTAMENTOS: (departamento, edificio, despacho, telefono, fax)

PERSONAL: (departamento, nif, primerApellido, segundoApellido, nombre, profesor)

2.9.6 Relaciones recursivas

Se trata de relaciones entre las entidades de un mismo conjunto de entidades, como en el ejemplo de la relación *SUPERVISA*, en el que a cada entidad *empleado* se le asocia otra entidad *empleado* del mismo conjunto de entidades, indicando que cada empleado es supervisado por otro empleado del mismo conjunto de entidades.



En este tipo de relaciones tiene una importancia fundamental el papel de las entidades en la relación.

El paso a tablas se hace como si se tratase de un conjunto de relaciones normal.

- Si es una relación 1:M (por ejemplo, un empleado supervisa a varios empleados, y un empleado sólo tiene un supervisor) se crea una tabla para el conjunto de entidades que participa como muchos en la relación, y se le añade la clave primaria del conjunto entidades que participa como uno, cambiándole el nombre.

<i>nombreEmp</i>	<i>dniEmp</i>	<i>telefono</i>	<i>dniSupervisor</i>
<i>Pepe</i>	<i>1</i>	<i>223344</i>	<i>3</i>
<i>Juan</i>	<i>2</i>	<i>224455</i>	<i>3</i>
<i>Luisa</i>	<i>3</i>	<i>226655</i>	

- Si es una relación M:N (por ejemplo, componentes de un ordenador), se crea una tabla para el conjunto de entidades y otra tabla para la relación con las claves primarias del conjunto de entidades.

<i>producto</i>	<i>codigo</i>
<i>Pentium Alfa</i>	<i>1</i>
<i>Monitor A</i>	<i>2</i>
<i>Monitor B</i>	<i>3</i>
<i>Disco duro A</i>	<i>4</i>
<i>Disco duro B</i>	<i>5</i>
<i>Pentium Beta</i>	<i>6</i>

<i>codigo1</i>	<i>codigo2</i>
<i>1</i>	<i>2</i>
<i>1</i>	<i>4</i>
<i>6</i>	<i>2</i>
<i>6</i>	<i>4</i>

2.10. Modelo Entidad-Relación Extendido

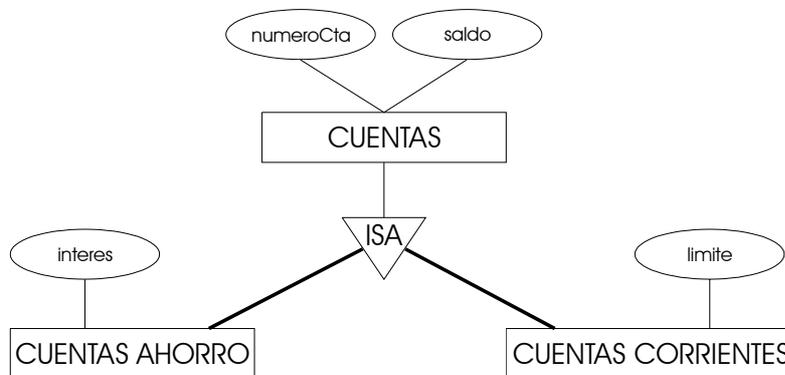
2.10.1. Generalización

Se trata de una abstracción de alto nivel en la que se pretenden resaltar las semejanzas. Se trata de una relación de inclusión que existe entre un conjunto de

entidades de nivel más alto y uno o más conjuntos de entidades en un nivel más bajo, en la que la diferencia entre los conjuntos de entidades se hace por herencia de atributos.

En el ejemplo del sistema bancario, el conjunto de entidades cuenta tiene los atributos *numeroCta* y *saldo*. Pero ocurre que las cuentas de ahorro y las cuentas corriente se pueden agrupar en un conjunto de entidades de nivel superior denominada cuentas. Cada una de estas entidades se describe mediante una serie de atributos que incluye a todos los atributos de cuenta más algunos atributos adicionales, como son las tasa de interés para el conjunto de entidades cuenta de ahorro y el límite del descubierto en una cuenta corriente. Debido a la similitud que existe entre estos dos conjuntos de entidades, se crea un conjunto de entidades de más alto nivel por generalización.

En el diagrama Entidad-Relación, se representa mediante un triángulo rotulado con "ISA" (es un) y líneas gruesas de la forma siguiente:



La transformación a tablas de una generalización en un diagrama E-R se realiza creando una tabla para el conjunto de entidades de más alto nivel y una tabla para cada uno de los conjuntos de entidades de más bajo nivel que contengan sus propios atributos más la clave primaria del conjunto de entidades de más alto nivel.

En el ejemplo de las cuentas, obtendríamos estas tres tablas:

<i>numeroCta</i>	<i>saldo</i>
1	100
2	200
3	300
4	400

<i>numeroCta</i>	<i>interes</i>
1	4,5
3	4

<i>NumeroCta</i>	<i>limite</i>
2	-100
4	0

2.10.2. Especialización

Se trata de una abstracción a alto nivel en la que se pretenden resaltar las diferencias, en la que se generan uno o más conjuntos de entidades de bajo nivel a partir de un conjunto de entidades de alto nivel.

La diferencia con la generalización es que con la generalización se construye un conjunto de entidades a partir de varios conjuntos de entidades, y por tanto todas las entidades de bajo nivel se encuentran también en el conjunto de entidades de alto nivel. Sin embargo, en el caso de la especialización puede ocurrir que al intentar dividir un conjunto de entidades en varios conjuntos de entidades, un grupo de entidades no pertenezca a ninguno de los conjuntos de entidades de nivel inferior.

Por ejemplo, un conjunto de entidades *PUBLICACIONES* se podría especializar en dos conjuntos de entidades *LIBROS* y *REVISTAS*, pero no todas las publicaciones tienen por que ser o un libro o una revista, ya que podrían ser un Trabajo de fin de carrera, un informe técnico, La solución no estará en crear conjuntos de entidades de nivel inferior para estas entidades, porque puede que no tengan más atributos que las distinguen.

Los atributos de los conjuntos de entidades de nivel inferior, al igual que en el caso de la generalización, se heredan.

La representación en el diagrama Entidad-Relación es la misma que en la generalización, salvo que no se utilizan líneas gruesas.

El paso a tablas de una especialización de un diagrama E-R se realiza de la misma forma que en la generalización, es decir una tabla para cada conjunto de entidades, en el que la tabla del conjunto de entidades de más alto nivel contiene todos sus atributos, y las de los conjuntos de entidades de más bajo nivel contienen todos sus atributos junto con los atributos que forman la clave primaria del conjunto de entidades.

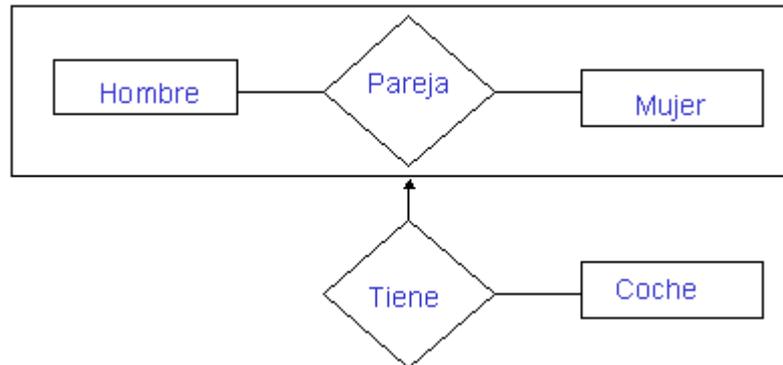
2.10.3. Agregación

La agregación se utiliza cuando un conjunto de relaciones entre entidades constituyen en sí un conjunto de entidades, que tienen su propio conjunto de atributos (lo que puede sonar a atributos descriptivos de una relación), y se relacionan con otro conjunto de entidades.

Supongamos ejemplo de dos conjuntos de entidades denominados *HOMBRE* y *MUJER* que tuviesen sus propios conjuntos de atributos y que están relacionados mediante el conjunto de relaciones *PAREJA*. Está claro que esta relación constituye en realidad una entidad, la entidad *PAREJA*, que tiene sus propios atributos como son *fechaAniversario* y *numeroDeHijos*. Ahora este nuevo conjunto de entidades se podría relacionar con otros conjuntos de entidades, como *VIVIENDAS*, *COCHES*, etc.

Cabría pensar que una agregación se podría resolver mediante una relación ternaria, pero veamos que esto no es así con un contraejemplo. Si los conjuntos de entidades *HOMBRE*, *MUJER* y *COCHE* estuviesen relacionados con una relación ternaria, sólo existirían en el conjunto de relaciones las parejas que tuviesen algún coche, lo cual no es cierto, porque puede haber parejas (relaciones entre *HOMBRE* y *MUJER*) que no tuviesen coche.

La agregación se representa con un recuadro que rodea a los conjuntos de entidades y el conjunto de relaciones, o bien con recuadro sobre el rombo del conjunto de relaciones, y se le suele denominar con un nombre.



La transformación de una agregación de un diagrama E-R en tablas es directa, donde se crea una tabla para cada conjunto de entidades y una tabla para cada conjunto de relaciones, excepto cuando se trata de relaciones 1:M, en las que se aplicarán las reglas ya conocidas.

HOMBRE (*idHombre*, ...)

MUJER (*idMujer*, ...)

PAREJA (*idHombre*, *idMujer*, ...)

COCHE (*idCoche*, ...)

TIENE (*idMujer*, *idHombre*, *idCoche*)

2.11. Diseño de bases de datos con diagramas de clases UML

El lenguaje de modelado unificado (UML) ofrece una herramienta gráfica para modelar varios componentes de un sistema software. Dentro de sus componentes, tenemos los *diagramas de clases*, *diagramas de casos de uso*, *diagramas de actividad* y *diagramas de implementación*.

Para el diseño de bases de datos, podemos utilizar los diagramas de clases, que muestran la estructura de los datos del sistema. A diferencia del modelo E-R, UML modela realmente objetos, que ofrecen métodos, y no sólo entidades.

Podemos describir los constructores básicos de diagramas UML en función de sus equivalentes en el modelo E-R.

UML muestra los objetos conjuntos de entidades (clases de objetos) como cuadros, incluyendo los atributos (y métodos) dentro del cuadro.

Los conjuntos de relaciones binarias se representan en UML con una línea que conecta las clases relacionadas, escribiendo su nombre junto a la línea. También se puede especificar el papel de cada una de las clases.

Si un conjunto de relaciones tiene atributos descriptivos, se crea una nueva clase con el nombre del conjunto y los atributos descriptivos, conectándola con una línea discontinua a la línea que conecta las clases relacionadas. Esta nueva clase podrá tratarse entonces como una clase más, como una agregación en el modelo E-R.

Las relaciones de orden mayor que dos se tienen que reducir a relaciones binarias, creando una nueva clase que representa la relación.

Las restricciones de cardinalidad se expresan con el formato *min..max*, donde *min* representa el mínimo número de relaciones en las que puede participar un objeto, y *max* representa el máximo, teniendo en cuenta que su significado es el inverso del representado en los diagramas E-R (representa la cardinalidad de entrada).

La generalización y especialización se modelan mediante la jerarquía de clases, conectando las clases de nivel inferior con una línea con punta de flecha hacia la clase de nivel superior. Existe también la posibilidad de expresar generalización *disjunta*, uniendo las clases del nivel inferior con una línea, conectada mediante otra línea con punta de flecha hacia la clase de nivel superior.

