

# INICIACIÓN A LAS BASES DE DATOS RELACIONALES CON ACCESS Y POSTGRESQL

24 horas.

6 sesiones formativas.

Días 15, 18, 21, 25, 27 y 29 de noviembre de 9.00 a 13.00 h.

Formador: Juan Pedro Pérez Alcántara.



Es el  
momento de ser  
**mucho más**  
globales  
humanos  
productivos  
actuales  
creativos

## Cuatro consejos para abordar el modelado

**Uno:** No se puede modelar lo que no se conoce.

**Dos:** No existe una **solución única** para un problema de modelado, tan sólo existen soluciones convenientes.

**Tres:** El problema nunca se soluciona a la primera, es un **proceso iterativo, reiterativo y evolutivo** que puede llegar a ser muy costoso en el tiempo. Se trabaja en el modelo, se somete a prueba, se corrige, se vuelve a probar... iterativamente. El problema ha de ser discutido con especialistas diferentes y visto desde todos los puntos de vista que se consideren necesarios. Es una labor de equipo. No se puede modelar con prisa. Es un proceso que hay que hacer con reposo.

**Cuatro:** Hay que ser **ordenado**. Hay que darle un nombre bien descriptivo y significativo a los componentes del modelo y documentarlo profusamente. Debe ser elegante, legible y sencillo, hasta donde el problema lo permita, claro.

## Problemas principales en el modelado

Un modelo de base de datos relacional ha de estar **NORMALIZADO**.

La normalización es el cumplimiento de una serie de reglas por parte del modelo de datos destinadas a erradicar los siguientes dos peligros potenciales que arruinarían la efectividad del modelo.

- **Falta de atomización:** La base de datos debe contener hechos lo suficientemente atomizados para la escala del problema en cuestión. Esto garantiza una gran versatilidad de la estructura a la hora de producir información.
- **Redundancia:** Los datos (especialmente los de naturaleza más compleja, como el texto) deben ser almacenados en la base de datos en un solo lugar, y no en varios. Repetir, redundar los datos es una forma segura de conseguir una base de datos inconsistente, inexacta, ineficaz e inestable, con el peligro de generar respuestas no veraces.

## Metodología para el diseño de un modelo de datos relacional

Existen multitud de **metodologías de modelado**.

Usaremos un método **iterativo de refinamiento por pasos**, basado en la metodología de modelado de entidad-relación, en la que aplicaremos una serie de normas.

- **Paso 1: comprensión del problema:** a efectos prácticos, muchas reuniones, mucho pensar y mucho papel garabateado.
- **Paso 2: identificación de los elementos del sistema y sus relaciones:** a partir del paso 1, **análisis y planteamiento inicial** de los componentes básicos del sistema descrito.
- **Paso 3: iteraciones de normalización:** el sistema es sometido a **sucesivos ciclos de estudio, análisis y cambios** hasta alcanzar un estado plenamente normalizado en el que todas las necesidades de información son satisfechas.

## Paso 1: Comprensión del problema

Sin comprender el problema no hay nada que modelar. Hay que describir el **problema**, los **procesos** y los **elementos del sistema** lo más detalladamente posible. Una técnica fundamental es escribirlo en papel, descriptivamente, exponiendo literalmente el problema. Esta descripción literal surge de un proceso de discusión con los especialistas temáticos.

De este proceso de discusión se obtienen los siguientes productos, llamados en su conjunto **Documentos de diseño**:

- una **descripción de los elementos, flujos, relaciones entre elementos y procesos** que se dan dentro del sistema que se quiere modelar;
- un documento en el que se perfilan las **necesidades de información** que debe satisfacer el sistema, es decir, la lista de preguntas mínimas que el sistema debe ser capaz de responder de forma veraz.

## Paso 2: Identificación de entidades y relaciones del sistema

El modelado entidad-relación precisa de la identificación de sus tres elementos principales:

- **entidades:** componentes del sistema;
- **atributos de estas entidades:** características que definen las entidades;
- **relaciones:** interacciones entre componentes en el sistema.

Una buena forma de comenzar a identificar estos elementos es hacerlo con un proceso semántico a partir de la descripción del sistema realizada en el paso 1. Según este proceso semántico:

- los **sustantivos** se corresponden con las entidades del modelo y sus atributos;
- los **verbos** se corresponden con relaciones del modelo.

## Paso 3: Iteraciones de normalización

- El diseñador plantea una **primera propuesta de modelo**, compuesta por **entidades (tablas)** con atributos (**campos**) y **relaciones** entre dichas entidades.
- Este modelo inicial es **analizado entidad por entidad, atributo por atributo**, en busca de posibles fuentes de **desnormalización** (redundancia de datos, falta de atomización, etc.).
- Si se encuentra una **potencial causa de desnormalización**, se arregla **atomizando** aún más las entidades, separándolas en entidades diferentes, y se vuelven a **replantear** los atributos de dichas entidades.
- Se comprueba si el sistema es capaz de satisfacer las **necesidades de información**.

Este proceso se repite iterativamente, refinando el modelo, hasta que se considere que:

- no se identifican posibles **fuentes de desnormalización** en el modelo;
- se satisfagan las **necesidades de información** planteadas.

## Un ejemplo clásico: La biblioteca

Para ejemplificar las dificultades propias del **proceso de creación** de un sistema de información computacional, vamos a estudiar la evolución de un caso de uso típico: la creación de un **sistema de información** para el control del inventario y la actividad de una pequeña biblioteca.

El diseñador decide hacerlo con una **base de datos relacional**, en este caso **Access**, ya que el sistema sólo va a ser manejado por una sólo persona a la vez.

En la biblioteca, actualmente, ya existe un sistema de información: un fichero con **tarjetas de cartón**. Por lo tanto, lo primero que deben hacer es comprender muy bien como funciona actualmente la biblioteca.



## Un ejemplo clásico: La biblioteca

Existe una **tarjeta de cartón por cada ejemplar** de un libro determinado. En cada tarjeta se consigna el nombre del libro, un código de copia (que cada libro lleva impreso en su solapa), los autores, la editorial y el ISBN. Por lo tanto, si hay dos Quijotes de la edición 2001 de la Editorial SM, existirán dos tarjetas idénticas salvo en el código de copia.

Cuando llegan nuevos **fondos bibliográficos**, los bibliotecarios cogen tantas tarjetas como ejemplares y rellenan los datos. Esto, para empezar, ya ha generado algunos problemas. Por ejemplo, los dos Quijotes anteriores poseen las siguientes tarjetas:

Nombre: El Quijote  
Autor: Cervantes  
Editorial: SM  
ISBN: 23-123-342132

Copia: 1

← Criterio dispar: inconsistencia →

← Error: inconsistencia →

Nombre: Don Quijote de La Mancha  
Autor: Miguel de Cervantes Saavedra  
Editorial: Santillana  
ISBN: 23-123-342132

Copia: 2

## Un ejemplo clásico: La biblioteca

La mayoría de las personas podrán identificar el error anterior y asumir que son dos copias del mismo libro, gracias a nuestra **capacidad de pensamiento en lógica difusa**. La lógica difusa hace a nuestro cerebro resistente a la información parcialmente incorrecta o incomplete.

Estas tarjetas están a disposición del público, **ordenadas alfabéticamente** en archivadores, para que se pueda consultar los fondos bibliográficos. Cuando un usuario quiere pedir prestado un libro, lo busca en el archivador. Si encuentra la tarjeta, quiere decir que está disponible para el préstamo. Retira la tarjeta, la entrega a los bibliotecarios y enseña su carnet de la biblioteca. Los bibliotecarios entonces se quedan con la tarjeta, le enganchan con un clip un papel en el que se indica el usuario que se ha llevado el libro y la fecha de devolución.

Este sistema de información no digital permite suplir los siguientes **requerimientos de información**:

- qué libros están disponibles en el momento para su préstamo;
- quién ha pedido prestado el libro, y cuando lo debe devolver.

## Un ejemplo clásico: La biblioteca

Sin embargo, el responsable de la biblioteca se queja de que **existen preguntas que o bien no tienen respuesta o bien tienen una respuesta costosa de obtener**. Por ejemplo:

- los usuarios **no pueden saber si un libro está en la biblioteca** si todas sus copias están en préstamo. Tendrían que acceder a las tarjetas de libros prestados y esto va contra las normas. Se podría solucionar con un listado adicional de todos los libros de la biblioteca, pero ya habría que manejar dos inventarios duplicados: etiquetas y lista. Esto es redundante y, por tanto, sujeto a errores de discrepancia de datos;
- no se tiene un **histórico de préstamos**, tan sólo se puede saber si un libro está disponible o no en el momento, pero no si lo estaba el lunes pasado. Una vez más, se puede llevar un registro paralelo de los préstamos, pero es redundante y muy propenso a errores;
- es **difícil controlar cuantos libros tiene prestado un usuario**, o qué copias de un libro están prestadas, ya que una vez el libro es prestado, la tarjeta de éste se traslada al fichero de préstamos, donde o bien son ordenados por usuario o por el propio libro. Se puede hacer dos copias de la tarjeta, una para ser ordenada por usuario y otra ordenada según el libro, pero vuelve a ser redundante.

## Un ejemplo clásico: La biblioteca

Los diseñadores deciden crear una tabla en Access preliminar donde consignar la información de los libros. Su primera idea es la siguiente:

Nombre	Copia	Autor	Editorial	ISBN	Prestado a	Devolución
El Quijote	1	Cervantes	SM	23-123-22123		
Don Quijote de la Mancha	2	Miguel de Cervantes	Santillana	23-123-22123		
La sombra sobre Innsmouth	1	Howard Phillips Lovecraft	Atlas	12-213-22132		
El horror de Dunwich	1	H.P. Lovecraft	Editorial Atlas Terror	12-213-32321		
Antología del cuento de terror	1	HP Lovecraft, August Derleth, Robert Bloch	Atlas Terror	12-213-22111		
Nueva antología del cuento de terror	1	August Derleth, Edgar Allan Poe, Howard P. Lovecraft	Atlas Colección Terror	12-213-11232		

Existen en esta primera versión varios problemas:

- **falta de criterio e inconsistencias en los datos:** *Cervantes* y *Miguel de Cervantes* hacen referencia al mismo autor, pero la máquina es incapaz de establecer esta relación puesto que el texto para él no significa gran cosa (por no decir nada);
- **inconsistencias en las copias:** los dos Quijotes son copias de un mismo libro pero, sin embargo, sus datos son inconsistentes. ¿Cuáles serán correctos, los de la copia 1 o los de la 2?

## Un ejemplo clásico: La biblioteca

Nombre	Copia	Autor	Editorial	ISBN	Prestado a	Devolución
El Quijote	1	Cervantes	SM	23-123-22123		
Don Quijote de la Mancha	2	Miguel de Cervantes	Santillana	23-123-22123		
La sombra sobre Innsmouth	1	Howard Phillips Lovecraft	Atlas	12-213-22132		
El horror de Dunwich	1	H.P. Lovecraft	Editorial Atlas Terror	12-213-32321		
Antología del cuento de terror	1	HP Lovecraft, August Derleth, Robert Bloch	Atlas Terror	12-213-22111		
Nueva antología del cuento de terror	1	August Derleth, Edgar Allan Poe, Howard P. Lovecraft	Atlas Colección Terror	12-213-11232		

Más problemas:

- **campos con múltiples datos atómicos:** el campo *Autor* contiene en **muchos casos a varios autores**. Esto crea problemas de búsqueda y ordenación. Por ejemplo, si buscamos todos los libros de HP Lovecraft ordenándolos alfabéticamente según su autor, *Nueva antología del cuento de terror* no aparecerá, ya que HP Lovecraft no es el primer autor de la lista de autores;
- **filtros imposibles:** podríamos buscar los libros de HP Lovecraft buscando la palabra *Lovecraft* en el campo de autores con un filtro. En ese caso, la falta de criterio a la hora de codificar el nombre de los autores hará que *La sombra sobre Innsmouth* no aparezca, ya que está mal escrito. Además, si existiera otro autor llamado *Edward Lovecraft*, también aparecería en el filtro. Si filtramos por el nombre completo, *HP Lovecraft*, el filtro tampoco será eficaz, ya que el nombre de Lovecraft ha sido escrito de múltiples formas.

## Un ejemplo clásico: La biblioteca

Se puede argüir que estos problemas pueden ser solventados con unas **normas y unas pautas bien establecidas** que hagan que los bibliotecarios introduzcan datos en el sistema con la calidad y el cuidado necesarios. Esto nos lleva a otro principio del diseño de sistemas de información: el **efecto del manual de instrucciones**:

*“Cuanto más abultado y extenso es el manual de instrucciones necesario para introducir datos y manejar un sistema de información, todo indica que peor hecho está”.*

Por ejemplo, en el manual de los bibliotecarios deberíamos consignar las **siguientes normas**:

- *antes de introducir un nuevo autor o editorial en el sistema, hay que buscar si previamente ya existe en él, examinar si todos están escritos de forma uniforme y escribirlo tal y como ya está presente en el sistema;*
- *si se reciben más copias de libros que ya están presentes en los fondos bibliográficos, se deben buscar en el sistema todas sus copias y asegurarse de que sus datos son iguales y correctos para todas ellas. Acto seguido, se podrán introducir las nuevas copias en el sistema, prestando siempre mucha atención a que no se repitan en las nuevas copias números de copia ya existentes previamente.*

## Un ejemplo clásico: La biblioteca

El director de la biblioteca decide que se necesita codificar nuevos datos sobre los autores y las editoriales, a saber:

- fecha de nacimiento y muerte de los autores;
- sede social de la editorial.

Por lo tanto, se introducen nuevos campos en la tabla para acomodar esta información.

Nombre	Copia	Autor	Fecha nacimiento autor	Fecha muerte autor	Editorial	Sede social editorial	ISBN
El Quijote	1	Cervantes	1547?	1616	SM	Madrid	23-123-22123
Don Quijote de la Mancha	2	Miguel de Cervantes	1548	1616	Santillana	Madrid	23-123-22123
La sombra sobre Innsmouth	1	Howard Phillips Lovecraft	1889	1937	Atlas	Badalona	12-213-22132
El horror de Dunwich	1	H.P. Lovecraft	1889	1937	Editorial Atlas Terror	Barcelona	12-213-32321
Antología del cuento de terror	1	HP Lovecraft, August Derleth, Robert Bloch	1889, 1909, 1917	1937, 1971, 1994	Atlas Terror	Barcelona	12-213-22111
Nueva antología del cuento de terror	1	August Derleth, Edgar Allan Poe, Howard P. Lovecraft	1909, 1809, 1889	1937, 1849, 1937	Atlas Colección Terror	Barcelona	12-213-11232

## Un ejemplo clásico: La biblioteca

Nombre	Copia	Autor	Fecha nacimiento autor	Fecha muerte autor	Editorial	Sede social editorial	ISBN
El Quijote	1	Cervantes	1547?	1616	SM	Madrid	23-123-22123
Don Quijote de la Mancha	2	Miguel de Cervantes	1548	1616	Santillana	Madrid	23-123-22123
La sombra sobre Innsmouth	1	Howard Phillips Lovecraft	1889	1937	Atlas	Badalona	12-213-22132
El horror de Dunwich	1	H.P. Lovecraft	1889	1937	Editorial Atlas Terror	Barcelona	12-213-32321
Antología del cuento de terror	1	HP Lovecraft, August Derleth, Robert Bloch	1889, 1909, 1917	1937, 1971, 1994	Atlas Terror	Barcelona	12-213-22111
Nueva antología del cuento de terror	1	August Derleth, Edgar Allan Poe, Howard P. Lovecraft	1909, 1809, 1889	1937, 1849, 1937	Atlas Colección Terror	Barcelona	12-213-11232

Esta solución, altamente insatisfactoria, presenta **nuevos problemas**:

- el sistema será incapaz de encontrar cual es la fecha de nacimiento o muerte de un autor cuando hay varios autores para un libro.



## Un ejemplo clásico: La biblioteca

Nombre	Copia	Autor	Fecha nacimiento autor	Fecha muerte autor	Editorial	Sede social editorial	ISBN
El Quijote	1	Cervantes	1547?	1616	SM	Madrid	23-123-22123
Don Quijote de la Mancha	2	Miguel de Cervantes	1548	1616	Santillana	Madrid	23-123-22123
La sombra sobre Innsmouth	1	Howard Phillips Lovecraft	1889	1937	Atlas	Badalona	12-213-22132
El horror de Dunwich	1	H.P. Lovecraft	1889	1937	Editorial Atlas Terror	Barcelona	12-213-32321
Antología del cuento de terror	1	HP Lovecraft, August Derleth, Robert Bloch	1889, 1909, 1917	1937, 1971, 1994	Atlas Terror	Barcelona	12-213-22111
Nueva antología del cuento de terror	1	August Derleth, Edgar Allan Poe, Howard P. Lovecraft	1909, 1809, 1889	1937, 1849, 1937	Atlas Colección Terror	Barcelona	12-213-11232

- Imaginemos que se detecta que la fecha de nacimiento de HP Lovecraft es incorrecta (de hecho, nació en 1890). Evaluemos que significa esto: hay que buscar a mano todas las copias de libros que tienen entre sus autores a Lovecraft y cambiar manualmente todas las fechas. Y si Lovecraft apareciera como autor en 2000 copias de libros....

## Un ejemplo clásico: La biblioteca

Nombre	Copia	Autor	Fecha nacimiento autor	Fecha muerte autor	Editorial	Sede social editorial	ISBN
El Quijote	1	Cervantes	1547?	1616	SM	Madrid	23-123-22123
Don Quijote de la Mancha	2	Miguel de Cervantes	1548	1616	Santillana	Madrid	23-123-22123
La sombra sobre Innsmouth	1	Howard Phillips Lovecraft	1889	1937	Atlas	Badalona	12-213-22132
El horror de Dunwich	1	H.P. Lovecraft	1889	1937	Editorial Atlas Terror	Barcelona	12-213-32321
Antología del cuento de terror	1	HP Lovecraft, August Derleth, Robert Bloch	1889, 1909, 1917	1937, 1971, 1994	Atlas Terror	Barcelona	12-213-22111
Nueva antología del cuento de terror	1	August Derleth, Edgar Allan Poe, Howard P. Lovecraft	1909, 1809, 1889	1937, 1849, 1937	Atlas Colección Terror	Barcelona	12-213-11232

- Una vez más, se produce una **redundancia** en los datos de las editoriales que puede dar lugar a inconsistencias: el bibliotecario puede equivocarse y codificar que para una copia de un libro de la editorial su sede es Barcelona, mientras que en otro consta que es Badalona... ¿cual es el dato correcto?

## Un ejemplo clásico: La biblioteca

Los diseñadores vuelven a la mesa de diseño y definen bien el problema, creando los **documentos de diseño** pertinentes.

*"En la biblioteca existen libros. Estos libros pueden tener varias copias. Cada copia tiene un código interno de la biblioteca. Sobre los libros se quiere saber sus autores, sus copias, su editorial, la sede de dicha editorial, su nombre y su ISBN. Sobre los autores, además, se quieren saber sus fechas de nacimiento y defunción. La biblioteca tiene un grupo de usuarios registrados. Estos usuarios pueden llevarse prestados libros, y hay que saber cuando fue realizado el préstamo y cuando se ha de efectuar la devolución. De los usuarios se quiere saber el nombre."*

La base de datos debería **ser capaz de contestar** en todo momento las siguientes cuestiones:

- proporcionar el catálogo completo de los fondos de la biblioteca;
- proporcionar la lista de usuarios;
- indicar qué libros están disponibles y cuales están prestados;
- indicar qué préstamos están cercanos al vencimiento;
- indicar los libros almacenados escritos por un autor;
- confeccionar una lista de libros editados por ciudades.

## Un ejemplo clásico: La biblioteca

A partir de la documentación anterior, se llega a las siguientes **conclusiones**:

- a primera vista, el sistema biblioteca se compone de los siguientes **componentes básicos**: libros y usuarios;
- en el sistema biblioteca se identifica la siguiente **relación básica**: los libros están relacionados con los usuarios porque hay un proceso llamado préstamo que los involucra a ambos.

Las entidades tienen **características básicas**, de las que el experto ha identificado:

- para los **libros**: ISBN, nombre, autores, copias y editorial;
- para los **usuarios**: nombre.

El **préstamo**, que es un **proceso del sistema** que relaciona a un libro con un usuario, también tiene sus propias características: tiene una fecha de devolución. Esto es de extrema importancia: no es una característica ni del libro ni del usuario, es del proceso que los une.

## Un ejemplo clásico: La biblioteca

Se toma la primera entidad identificada en el sistema, **libro**, y se estudia para modelarla. Las entidades del modelo tienen las siguientes características:

- las **entidades** son **conjuntos de objetos homogéneos** que tienen rasgos (llamados atributos o campos) comunes. Por ejemplo, "libro" y "río" son ejemplos de entidades;
- un **miembro concreto** de una entidad se denomina **registro**. Así, "El Quijote" es un registro de "libro" y "Guadalquivir" de "río".

Identificada la entidad libro ahora debe someterla al siguiente proceso:

- **identificación** de los atributos de la entidad;
- **evaluación** de la **normalización** de la entidad en su estado actual y corrección de los defectos que pueda presentar. Este paso se itera cuantas veces haga falta hasta llegar a una entidad que cumple todas las **reglas de normalización**.

El cumplimiento de las **reglas de normalización** nos asegura que el modelo es sólido frente a redundancia e inconsistencias en los datos.

## Un ejemplo clásico: La biblioteca

Se toma la primera tabla de libros hecha como punto de partida y se analiza su normalización, chequeando el cumplimiento de las reglas:

Nombre	Copia	Autor	Fecha nacimiento autor	Fecha muerte autor	Editorial	Sede social editorial	ISBN
El Quijote	1	Cervantes	1547?	1616	SM	Madrid	23-123-22123
Don Quijote de la Mancha	2	Miguel de Cervantes	1548	1616	Santillana	Madrid	23-123-22123
La sombra sobre Innsmouth	1	Howard Phillips Lovecraft	1889	1937	Atlas	Badalona	12-213-22132
El horror de Dunwich	1	H.P. Lovecraft	1889	1937	Editorial Atlas Terror	Barcelona	12-213-32321
Antología del cuento de terror	1	HP Lovecraft, August Derleth, Robert Bloch	1889, 1909, 1917	1937, 1971, 1994	Atlas Terror	Barcelona	12-213-22111
Nueva antología del cuento de terror	1	August Derleth, Edgar Allan Poe, Howard P. Lovecraft	1909, 1809, 1889	1937, 1849, 1937	Atlas Colección Terror	Barcelona	12-213-11232

## Un ejemplo clásico: La biblioteca

El campo autor presenta múltiples problemas a la hora de filtrar y asegurar la integridad de la información. Se propone una posible solución: poner varios campos para los autores.

Nombre	Copia	Autor1	Autor2	Autor3	Fecha nacimiento autor1	Fecha nacimiento autor2	Fecha nacimiento autor3	Prestado a
Antología del cuento de terror	1	HP Lovecraft	August Derleth	Robert Bloch	1889	1909	1917	

## Un ejemplo clásico: La biblioteca

Evidentemente, esta solución es **altamente insatisfactoria**, por múltiples motivos:

- no sólo hay que **dupli, tripli, cuatri ... n-replicar** los campos de autores, sino todos los campos que les afecten, como por ejemplo la fecha de nacimiento y muerte de los mismos;
- si tenemos la tabla preparada para 4 autores, aquellos libros que tienen **menos de 4** tendrán un montón de campos vacíos o nulos. Esto no es bueno;
- peor, si de repente aparece una revista con 30 autores... ¿meteremos en la tabla **90 nuevos campos**?;
- por no hablar de las **redundancias** que se producirían, una vez más, en los datos;
- los filtros y las **consultas** al sistema se hacen impracticables, ya que tendríamos que tener en cuenta para ello el número total de campos de autores y la posibilidad de que el autor buscado esté en cualquiera de ellos, para cualquiera de los libros... imposible.

A todos estos efectos se le conoce en su conjunto como la **aberración de la tabla de infinitos campos**, ya que la tabla tiende a crecer horizontalmente a medida que los datos que se introducen en ella se hacen cada vez más complejos. Solución inaceptable.



## Un ejemplo clásico: La biblioteca

A raíz de comprobar esta primera aberración en la normalización se detecta **otro problema más**.

Es evidente que los libros tienen autores y que los publican editoriales, y esto es una realidad del sistema que el modelo debe contemplar. Pero el lugar en el que se encuentran codificados los datos de estas realidades del sistema no es el adecuado. El problema estriba en que **estamos mezclando en una misma entidad componentes diferentes del sistema**.

Por fuerza, debemos codificar de alguna manera el hecho de qué autor o autores han realizado la obra o qué editoriales las publican, pero tampoco se debe obviar el hecho de que el director de la biblioteca dio la orden de que se tienen que recoger datos adicionales sobre estos dos componentes del sistema. Esto hace que, de alguna manera, el concepto de *Autor de un libro* y el de *Editorial* tome una fuerte, por decirlo de alguna manera, *personalidad propia* dentro de nuestro modelo, tome más importancia de la que tenía en un principio.

Cuando esto pasa, la **monolítica entidad Libro** que teníamos al principio debe **descomponerse en realidades más sencillas** que lo describen. Por lo tanto, de *Libro* el especialista va a extraer dos nuevas entidades: *Autor* y *Editorial*.

## Un ejemplo clásico: La biblioteca

Existe, además, una razón de peso para descomponer *Libro* en *Libro*, *Autor* y *Editorial*: la **redundancia**.

El estado actual de la tabla:

Nombre	Copia	Autor	Fecha nacimiento autor	Fecha muerte autor	Editorial	Sede social editorial	ISBN
El Quijote	1	Cervantes	1547?	1616	SM	Madrid	23-123-22123
Don Quijote de la Mancha	2	Miguel de Cervantes	1548	1616	Santillana	Madrid	23-123-22123

Nótese como, para un mismo autor, Miguel de Cervantes, su nombre se ha **introducido dos veces** en la base de datos, además, de forma incoherente y sin criterio. Nótese también como su fecha de nacimiento y defunción también han sido introducidas **dos veces**, con peligro de que sean incongruentes en caso de fallo y con la incomodidad de, si nuevas investigaciones dicen que Cervantes nació un año antes, tener que buscar TODAS las fechas de nacimiento de Cervantes y cambiarlas al nuevo valor.

Todo esto es **REDUNDANTE**. Los datos, como el nombre del autor, su fecha de nacimiento, etc. deben ser introducidos **UNA SOLA VEZ** en la base de datos. Para introducirlos una sola vez, la solución es independizar el concepto de *Autor* del concepto de *Libro*. Son conceptos relacionados, pero son realidades diferentes dentro del sistema.

## Un ejemplo clásico: La biblioteca

A la vista de todo lo anterior, enunciamos nuestra primera **causa de desnormalización** y tres reglas para evitarlo.

### **Causa 1: Múltiples atributos de una entidad hacen referencia al mismo dato**

Una entidad **no puede contener más de un atributo que haga referencia al mismo dato y con el mismo propósito**. Asimismo, en una entidad no deben de incluirse como atributos aquellos que sean propios de otras entidades.

**1ª Norma: nunca una entidad contendrá más de un atributo destinado a contener el mismo dato para el mismo propósito.** Por ejemplo, una entidad “Libro” no tendrá un atributo “Autor 1” y otro “Autor 2” para contener los nombres de sus dos hipotéticos autores. En su lugar, *Autor* se independizará como entidad por derecho propio de *Libro*. De esta norma se deriva la siguiente.

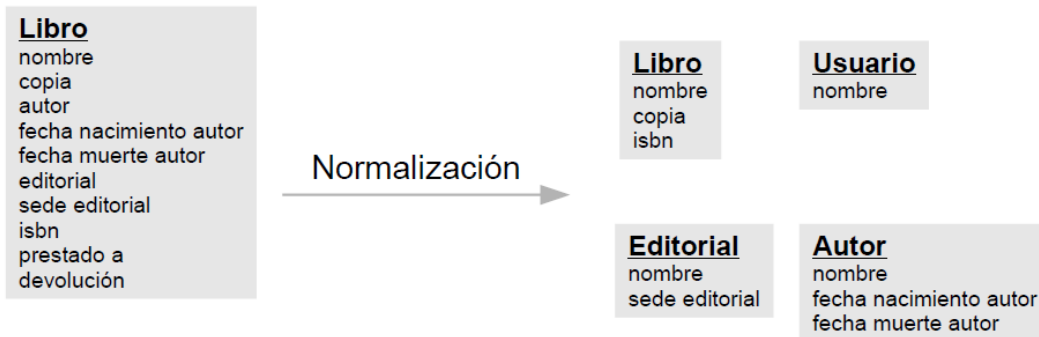
**2ª Norma: nunca se añadirán nuevos atributos a una entidad de un modelo de datos en un entorno de producción.** Por ejemplo, en caso de haber modelado mal los autores del objeto “Libro” según la propuesta de los bibliotecarios y encontrarnos con un libro con tres autores, tendríamos que crear un nuevo atributo llamado “Autor 3”. Esto está terminantemente prohibido. Si nos vemos en la necesidad de añadir campos a un sistema en producción, más nos vale repasarlo entero, porque seguro que algo falla.

## Un ejemplo clásico: La biblioteca

**3ª Norma: una entidad nunca contendrá atributos pertenecientes a otras entidades, sino una referencia a éstos.**

Por ejemplo, en la entidad *Libro* no debería aparecer unos atributos *Nombre de la editorial* y *Dirección de la editorial*, sino que dichos atributos deberían ser parte de una entidad *Editorial* bien diferenciada de (pero relacionada con) *Libro*.

Llegados a este paso, por tanto, el modelo ha evolucionado de la siguiente manera:



## Un ejemplo clásico: La biblioteca

Ahora se vuelve a iterar sobre las entidades del modelo, para ver si alguna de las **nuevas entidades surgidas de la anterior iteración de normalización presenta problemas:**

- **la entidad Usuario parece correcta.** Cada usuario tiene un nombre, y está normalizado puesto que cada usuario de la biblioteca estará codificado una sólo vez: un registro de la entidad usuario por cada usuario;
- **la entidad Autor también está normalizada:** cada autor será un registro en la tabla Autor. Un nombre, una fecha de nacimiento y otra de defunción. Todo codificado una sola vez;
- sin embargo, la **entidad Editorial si tiene problemas.** No hay problema con el nombre de la editorial, puesto que están consignados una sola vez. Sin embargo, si supone un problema la sede social, puesto que es un dato complejo que se repite una y otra vez si varias editoriales tienen la misma sede social. Esto es redundante y peligroso.

Editorial	Sede social editorial
Editorial SM	Madrid
Editorial Santillana	Madrid
Wiley & Sons	Nueva York
McGraw Hill	Nueva York

## Un ejemplo clásico: La biblioteca

Por lo tanto, la sede social de la editorial debe dejar de ser un atributo de la entidad *Editorial* para constituirse por derecho propio en una entidad:

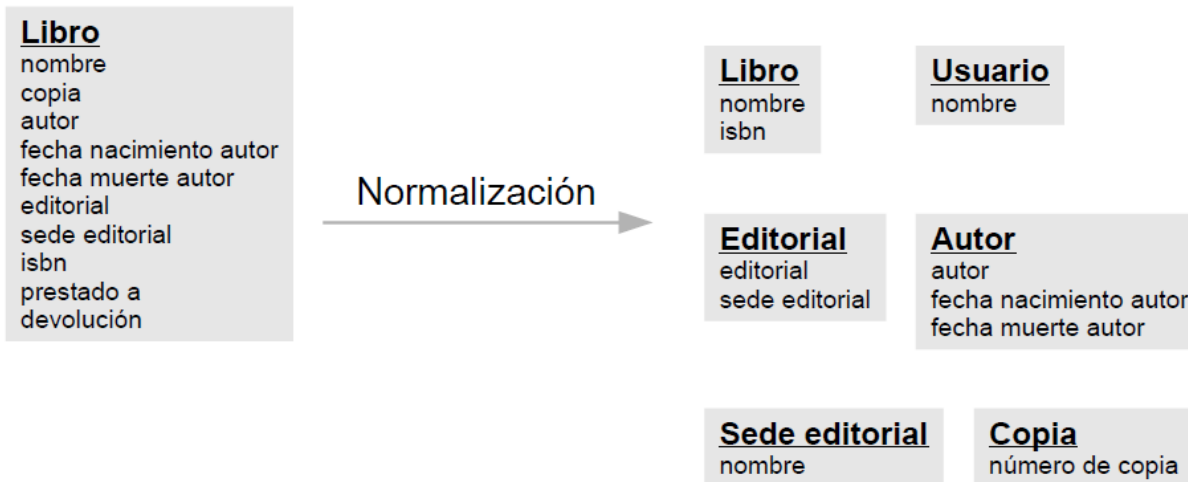
**Sede editorial**  
nombre

El problema serio, porque es un **problema conceptual**, viene con la entidad *Libro*. Si Libro se queda como está, obtendremos algo así:

Nombre	Copia	ISBN
Don Quijote de la Mancha	1	23-123-22123
Don Quijote de la Mancha	2	23-123-22123
Don Quijote de la Mancha	3	23-123-22123
Don Quijote de la Mancha	4	23-123-22123

Evidentemente, aquí hay problemas de **normalización**, puesto que se repiten los datos de nombre e ISBN, que son datos complejos con alto peligro de redundancia. Estos datos del libro deben ser introducidos en el sistema una sola vez. Por tanto, el concepto de *Copia* debe ser **disociado del concepto de Libro**, de forma que *Copia* y *Libro* deben ser entidades diferentes.

## Un ejemplo clásico: La biblioteca



## Un ejemplo clásico: La biblioteca

### Causa 2: Redundancia sistemática de datos complejos

La **repetición de datos complejos** (cadenas de caracteres, sobre todo) que no sean identificadores unívocos (claves foráneas) de registros de otras entidades en los atributos de una o varias entidades del modelo es un claro signo de no normalización.



## Un ejemplo clásico: La biblioteca

### Causa 2: Redundancia sistemática de datos complejos

**4ª Norma: nunca se creará un atributo en una entidad para contener una cadena de texto u otro tipo de dato complejo que sea susceptible de repetirse en varios de sus registros o en registros de otra entidad.** En su lugar, el conjunto de datos complejos deberá convertirse en una entidad con un identificador único que será referenciado en la primera entidad. Por ejemplo, la entidad “Libro” tiene un atributo “Autor”. Es incorrecto repetir la cadena de caracteres “Miguel de Cervantes” en varios registros de la instancia “Libro”. En su lugar, “Autor” debe convertirse en una instancia independiente de “Libro”, de modo que “Libro” contendría una referencia al registro “Miguel de Cervantes” de la entidad “Autor”.

## Un ejemplo clásico: La biblioteca

Por último, el problema inicial del atributo Autor de la primera propuesta de los bibliotecarios se evita de la siguiente manera:

### **Causa 3: Codificación de hechos atómicos de información mezclados en un atributo**

La inclusión de varios hechos atómicos en un mismo atributo de una entidad es un claro signo de no normalización.

## Un ejemplo clásico: La biblioteca

**5ª norma: nunca se creará un campo de texto destinado a contener palabras o cadenas de texto separadas por comas o cualquier otro carácter (/ , |, etc.).** Un campo debe contener un y sólo un átomo de información. En su lugar, el conjunto de datos atómicos será almacenado en una nueva entidad y la entidad original contendrá referencias a dicha entidad. Retomando el ejemplo de las entidades “Libro” y “Autor”, sería erróneo que la entidad “Libro” tuviera un atributo llamado “Autores” en el que se almacenara el nombre de los mismos, independientemente del número de autores. Sería erróneo encontrarse un registro de “Libro” que contuviera en un atributo “Autores”, por ejemplo, “Mike Mignola, Richard Corben”. En su lugar, “Autores” debe convertirse en una entidad propia del que “Mike Mignola” y “Richard Corben” serían registros, y el registro de libro citada al principio contendría una referencia a dichos registros.

## Relaciones entre entidades

Ahora que ya se tiene el catálogo completo de entidades del modelo, viene el siguiente paso en el proceso de modelado: establecer las relaciones entre las entidades.

Las relaciones del modelo tienen las siguientes características:

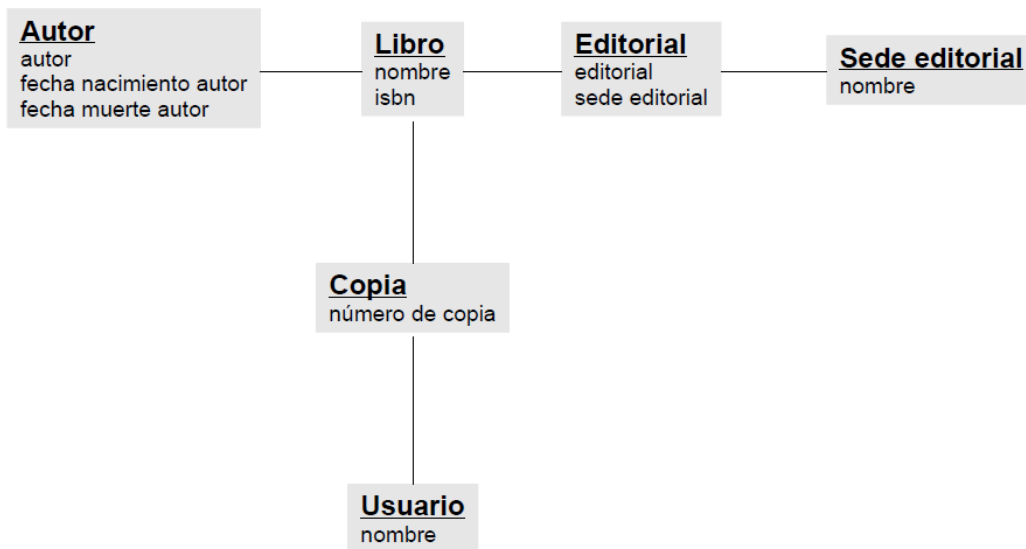
- **las relaciones se establecen entre dos (y sólo dos) entidades del modelo.** Cualquier relación percibida como a tres bandas o más puede ser descompuesta en relaciones sencillas dos a dos;
- **su principal propiedad es la cardinalidad:** con cuántos registros de la entidad B se relaciona un registro de la entidad A y viceversa.

## Relaciones entre entidades

La **cardinalidad** de una relación puede ser, a grandes rasgos, de tres tipos distintos:

- **uno a uno:** un registro de A se relaciona con un y sólo un registro de B. Por ejemplo, la relación existente entre el número de pasaporte de un país y sus ciudadanos: un ciudadano sólo puede tener un número, y ese número está asignado sólo a ese ciudadano;
- **uno a varios:** un registro de A se relaciona con varios registros de B. Por ejemplo, la relación "ser madre de/ser hijo de": un registro de A (una "madre") puede estar relacionado con varios registros de B (con varios "hijos"), pero un registro de B (un "hijo") sólo puede estar relacionado con un registro de A (con una "madre");
- **varios a varios:** un registro de A se relaciona con varios registros de B, y viceversa. Por ejemplo, la relación existente entre el conjunto de los libros escritos y el conjunto de los autores de libros: un registro de A (un "libro") puede estar asociado a muchos registros de B ("autores de libros"), ya que un libro puede tener más de un autor, mientras que un registro de B (un "autor") puede estar asociada con varios registros de A ("libros"), ya que un mismo autor puede escribir varias obras.

## Relaciones entre entidades



## Relaciones entre entidades

Las relaciones de nuestro modelo se describen literalmente de la siguiente manera. Hacer la descripción literal siempre es un buen ejercicio para tener bien claras las relaciones y analizarlas:

- los autores están relacionados con los libros ya que los autores escriben libros;
- los libros están relacionados con las editoriales ya que las editoriales editan libros;
- las editoriales están relacionadas con las sedes de las editoriales (que son ciudades) puesto que las editoriales tienen su sede en una ciudad.
- los libros están relacionados con sus copias puesto que la entidad libro describe los detalles del libro, mientras que la entidad copia hace recuento de las copias individuales de un libro dado;
- las copias están relacionadas con los usuarios puesto que son las copias de los libros, y no los libros en sí, los que son dados en préstamo a los usuarios.

## Relaciones entre entidades

Ahora hay que estudiar la **cardinalidad** de las relaciones descritas. El método es claro, y si se hace tal y como se describe, no debería haber problemas:

- cogemos una **relación**;
- empezando por una de las entidades relacionadas, nos preguntamos: **¿con cuántos registros de la entidad que está al otro lado de la relación está relacionado un (y solo un) registro de la entidad de origen?** Esta pregunta será contestada, básicamente, con un **1 o con un muchos**. Esta es la **cardinalidad parcial** de la entidad de origen elegida a la de destino. Por ejemplo, en la relación Libro - Copia, partiendo de libro, nos preguntaríamos: ¿con cuántas copias está relacionado un sólo libro? La respuesta es con muchas, ya que un libro puede tener muchas copias;
- ahora nos hacemos la misma pregunta, pero en **sentido inverso**. Eso nos dará la cardinalidad parcial, pero en la otra dirección. En el ejemplo anterior, nos preguntaríamos ¿con cuántos libros estará relacionada una copia? La respuesta es sólo con una, ya que una copia sólo puede ser copia de un libro concreto.



## Relaciones entre entidades

La **composición** de ambas cardinalidades parciales nos da la **cardinalidad final**:

- si ambas cardinalidades parciales son 1, la cardinalidad final es 1 a 1;
- si una cardinalidad parcial es 1 y la otra es muchos, la cardinalidad final es de 1 a muchos;
- si ambas cardinalidades parciales es muchos, la cardinalidad final es de muchos a muchos.

De esta forma:

- **autor-libro**: es N a N, es decir, muchos a muchos. Un autor puede haber escrito muchos libros, y un libro puede haber sido escrito por muchos autores;
- **libro-editorial**: es 1 a N, es decir, de uno a muchos, desde editoriales a libro, ya que un libro sólo puede estar publicado por una editorial, mientras que una editorial puede publicar muchos libros;
- **editorial - sede editorial**: es 1 a N, desde la sede a la editorial, ya que una editorial sólo puede tener una sede, mientras que una ciudad puede ser sede de muchas editoriales;

## Relaciones entre entidades

- **libro-copia:** es 1 a N, desde los libros a las copias, ya que un libro puede tener muchas copias, pero una copia sólo puede pertenecer a un libro;
- **copia-usuario:** es N a N, ya que una copia puede haber sido prestada (no simultáneamente, ahora veremos que significa esto) a muchos usuarios diferentes, mientras que un usuario puede haber recibido en préstamo más de una copia.

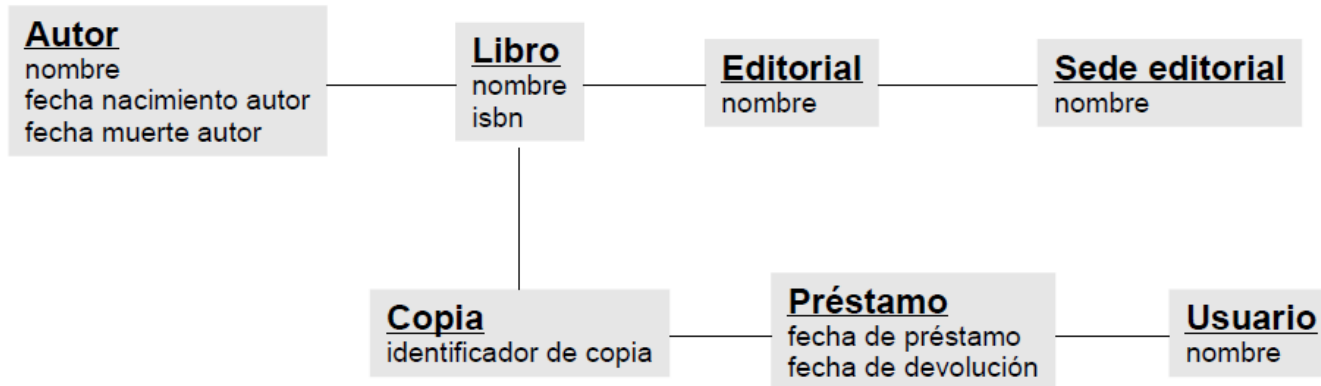
En este momento se puede considerar el modelo conceptual prácticamente terminado, pero vamos a reflexionar un poco sobre el concepto del **préstamo**.

El **préstamo** es el proceso de asignar una copia determinada de un libro a un usuario determinado. Como **proceso del sistema** que es, tiene sus propios atributos que describen el acto del préstamo. Por ejemplo, para satisfacer las necesidades del director de la biblioteca de obtener un histórico de préstamos, se puede consignar en el préstamo tanto la fecha del préstamo como la fecha de devolución.

## Relaciones entre entidades

Esto se refleja en el modelo conceptual de la siguiente forma:

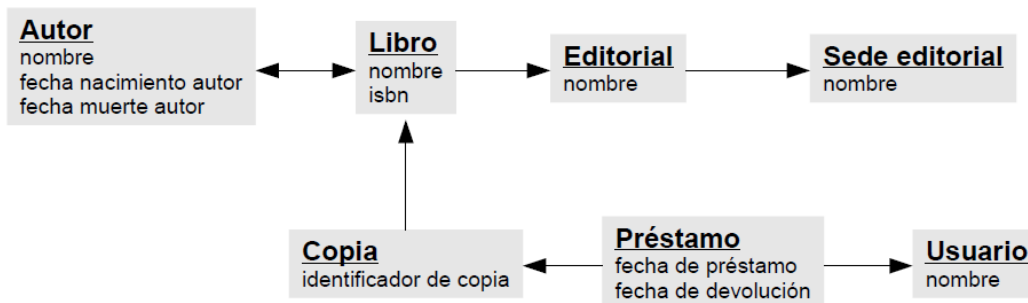
- el **proceso de préstamo es una entidad más del modelo**;
- sus **atributos** son: el usuario al que se le presta, la copia prestada, la fecha de préstamo y la fecha de devolución.



## Relaciones entre entidades

El último paso para terminar el modelo conceptual es simbolizar correctamente la cardinalidad de las relaciones:

- las relaciones 1 a 1 se simbolizan con un trazo sin flechas;
- las relaciones 1 a N se simbolizan con una flecha que va desde la entidad el lado del N al lado del 1;
- las relaciones N a N se simbolizan con una doble flecha.



## Paso del modelo conceptual al modelo físico

A partir del **modelo conceptual** se creará ahora un **modelo físico** que será el **conjunto de tablas con sus campos** que constituirá la base de datos real funcionando dentro de un SGBD. El paso al modelo físico es un proceso **mecánico** sencillo, pero hay que entender muy bien dos conceptos fundamentales: **la clave primaria y la clave foránea**.

Se ha dicho que una de las ventajas del modelado entidad-relación es que los datos establecen entre ellos **relaciones naturales**. Los datos están contenidos en tablas, por lo tanto, entre ellas se pueden establecer relaciones. Hemos identificado las relaciones de nuestro modelo conceptual, pero ¿cómo se establece dicha relación, físicamente, entre las tablas que forman la base de datos?

## Paso del modelo conceptual al modelo físico

Cada tabla debe tener un **atributo**, o **campo**, que sirva para **identificar de forma unívoca cada registro de dicha tabla**. Para ello, dicho campo debe cumplir, obligatoriamente, dos condiciones:

- que jamás pueda tener un **valor nulo**, es decir, sin dato (0 es un dato, por contra, nulo, también conocido como *null*, significa "ausencia de dato", que es bien distinto a 0);
- que cada valor contenido en dicho campo para cada registro sea **único**, diferente a cualquier otro valor de dicho campo para cualquier otro registro de la misma tabla.

## Paso del modelo conceptual al modelo físico

Por ejemplo, los libros tienen un ISBN único a nivel mundial que es una buena clave primaria, las personas de una misma nacionalidad tienen un número de pasaporte también único, etc.

Las claves primarias sirven, por tanto, para **identificar de forma unívoca** a todos los registros de una tabla. Todas las tablas de la base de datos tienen que tener una clave primaria, si no, son inútiles para el sistema.

## Paso del modelo conceptual al modelo físico

Las claves primarias también pueden estar compuestas por **múltiples campos** que, en su conjunto, pero no por separado, cumplen los requisitos exigidos a toda clave primaria. Por ejemplo, en España, cada municipio tiene un código compuesto de dos cifras que identifican su provincia y otras tres que identifican al municipio dentro de la provincia (el llamado *código INE*). El código compuesto es una buena clave primaria, pero mejor aún son los dos códigos por separado actuando como clave primaria múltiple (en verde oscuro, los campos claves primaria):

cod_ine		nombre	cod_ine_provincial		cod_ine_municipal	nombre
41001		Aguadulce	41		001	Aguadulce
41002		Alanís	41		002	Alanís



## Paso del modelo conceptual al modelo físico

Dado que las claves primarias identifican unívocamente a cada registro de cada tabla, para relacionar un grupo de registros de una tabla con otro grupo de registros de otra lo que se hace es que los registros de una tabla contengan campos que hagan referencia a los valores de las claves primarias de otras tablas. Estos campos que contienen valores que hacen referencia a valores que existen como claves primarias de otras tablas se llaman *campos claves foráneas* o *claves ajenas*.

Por ejemplo, provincias y municipios:

cod_ine_provincial	nombre
11	Cádiz
41	Sevilla

cod_ine_provincial	cod_ine_municipal	nombre
41	001	Aguadulce
41	002	Alanís
11	001	Alcalá de los Gazules
11	002	Alcalá del Valle

## Paso del modelo conceptual al modelo físico

cod_ine_provincial	nombre
11	Cádiz
41	Sevilla

cod_ine_provincial	cod_ine_municipal	nombre
41	001	Aguadulce
41	002	Alanís
11	001	Alcalá de los Gazules
11	002	Alcalá del Valle

La clave primaria de las provincias es *cod\_ine\_provincial*. Es único y nunca es nulo. La clave primaria de los municipios es múltiple, estando constituida por los campos *cod\_ine\_provincial* y *cod\_ine\_municipal*. Juntos, estos campos tienen las características de una clave primaria: no son nulos en su conjunto y son únicos pareja a pareja.

## Paso del modelo conceptual al modelo físico

<i>cod_ine_provincial</i>	nombre
11	Cádiz
41	Sevilla

<i>cod_ine_provincial</i>	<i>cod_ine_municipal</i>	nombre
41	001	Aguadulce
41	002	Alanís
11	001	Alcalá de los Gazules
11	002	Alcalá del Valle

Pero además, el campo *cod\_ine\_provincial* de los municipios es una clave foránea del campo homónimo de provincias, por lo que los valores presentes en el campo de la tabla de municipios son valores de la clave primaria de las provincias, por lo que, de facto, lo que se está produciendo es que estamos creando una relación entre algunos municipios y algunas provincias: aquellos municipios que tengan en su campo clave foránea *cod\_ine\_provincial* el valor del código de Sevilla estarán relacionados con dicha provincia, y análogamente con Cádiz. Por lo tanto, igualando clave primaria=clave foránea, obtenemos la siguiente información: Aguadulce y Alanís son municipios de la provincia de Sevilla, mientras que Alcalá de los Gazules y Alcalá del Valle son municipios de Cádiz.

## Paso del modelo conceptual al modelo físico

cod_ine_provincial	nombre
11	Cádiz
41	Sevilla

cod_ine_provincial	cod_ine_municipal	nombre
41	001	Aguadulce
41	002	Alanís
11	001	Alcalá de los Gazules
11	002	Alcalá del Valle

Esta es la base del establecimiento de relaciones entre entidades (tablas) del modelo relacional: campos clave primaria que se corresponden e igualan sus valores con campos clave foráneas de otras tablas. Por eso se dice que los registros de una tabla están relacionados o apuntan a los registros de alguna otra si la primera tabla posee una clave foránea a la clave primaria de la segunda.

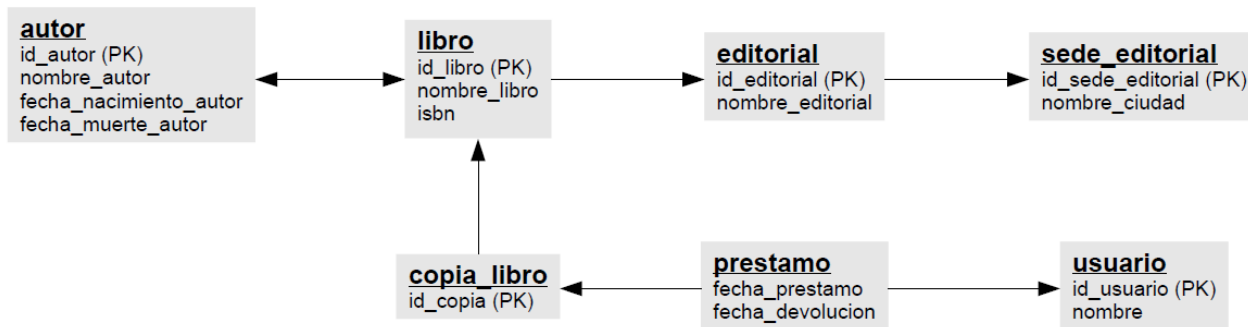
## Paso del modelo conceptual al modelo físico

Para pasar del modelo conceptual al modelo físico se deben seguir las siguientes **directrices**:

- todas **las entidades se convertirán en tablas**;
- todos los **atributos de las entidades se transformarán en campos**, de un **tipo** de terminado;
- todas las **tablas deben tener una clave primaria**. La clave primaria es un campo que identifica unívocamente al registro. Si ya existe un campo que cumpla en la actualidad y en el futuro la condición de unicidad, se puede usar como clave primaria. Si no, se le asignará a la tabla un nuevo campo único, creado de forma artificial para la ocasión. También puede haber claves primarias compuestas por varios campos.

## Paso del modelo conceptual al modelo físico

De esta manera, nuestras tablas quedan con los siguientes campos. Los campos clave primaria se denotan tradicionalmente con las siglas *PK* (Primary Key):



## Paso del modelo conceptual al modelo físico

Las relaciones también ejercen su efecto sobre el modelo:

- **en las relaciones 1 a 1**, cualquiera de las dos claves primarias de las entidades relacionadas viajará como clave foránea a la otra tabla;
- **en las relaciones 1 a N**, la clave primaria del lado del 1 viajará como clave foránea a la tabla del lado del N, es decir, la clave primaria se desplazará en el sentido contrario al de la flecha;
- **las relaciones N a N** se convertirán en una nueva tabla que existirá entre las dos entidades relacionadas, que se denomina tabla de relación, y que contiene como claves foráneas las dos claves primarias de las tablas relacionadas. En este contexto, lo que importa es la pareja de claves primarias relacionadas, y no cada una por separado. Las dos claves foráneas forman, de hecho, la clave primaria múltiple de la tabla de relación.

## Paso del modelo conceptual al modelo físico

Una vez realizado el trasvase entre tablas de claves foráneas, el modelo queda definitivamente confeccionado. Tradicionalmente, los campos clave foránea se denotan con *FK* (de Foreign Key).

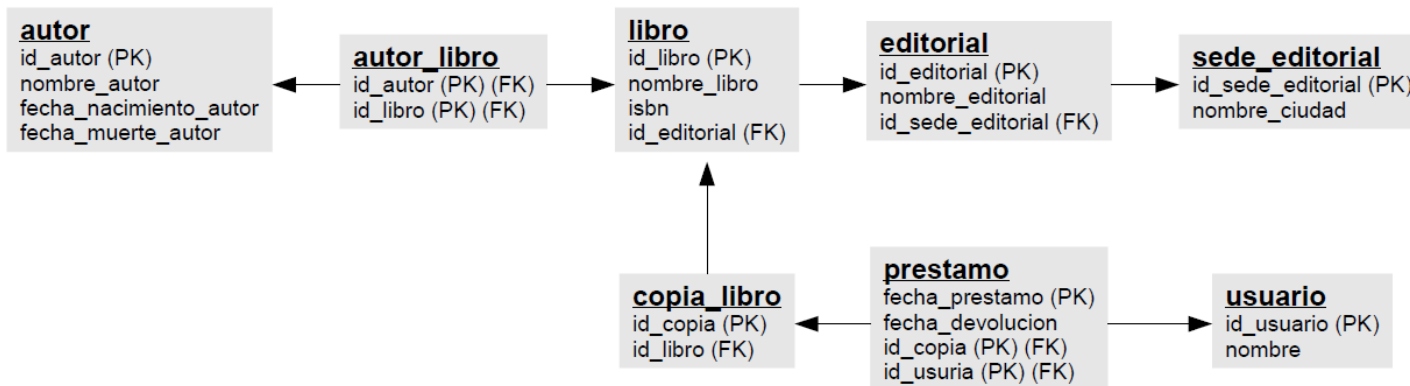




Tabla: autor			
id_autor	nombre_autor	fecha_nacimiento_autor	fecha_muerte_autor

1	Miguel de Cervantes	1548	1616
2	HP Lovecraft	f1	f2
3	Robert Bloch	f3	f4
4	August Derleth	f5	f6

Tabla: autor_libro	
id_autor	id_libro

1	1
2	2
2	3
3	3

Tabla: libro			
id_libro	nombre_libro	id_editorial	isbn

1	El Quijote	1	23-123-221231
2	El Horror de Dunwich	2	23-113-323434
3	Antología del cuento de terror	2	23-113-223212

Tabla: editorial		
id_editorial	nombre_editorial	id_sede_editorial

1	Editorial SM	1
2	Editorial Atlas	2

Tabla: copia_libro	
id_copia	id_libro

qui1	1
qui2	1
qui3	1
dun1	2
dun2	2
antt1	3
antt2	3
antt3	3

Tabla: usuario	
id_usuario	nombre_usuario

1	Fulano de Tal
2	Cetano de Cual

Tabla: prestamo			
id_copia	id_usuario	fecha_prestamo	fecha_devolucion

qui1	1	10/10/04	12/10/04
qui2	2	10/10/04	12/10/04

Tabla: sede_editorial	
id_sede_editorial	nombre_ciudad

1	Madrid
2	Barcelona