

Laboratorio 2

- Modelo relacional y normalización



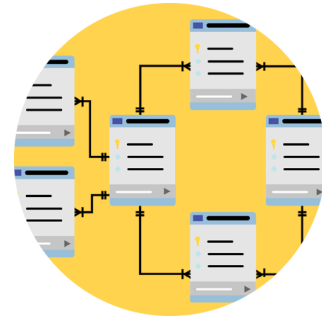
Modelo Relacional

Tras ser postulado por Edgar Frank Codd en 1970, el modelo relacional se estableció rápidamente como un paradigma importante en los modelos de bases de datos, actualmente, el más utilizado en la gestión de datos dinámicos.

La principal idea en la que se basa este modelo es el uso de **relaciones**. Consiste en el almacenamiento de datos en tablas compuestas por filas (**tuplas**) y columnas (**campos o atributos**). Es más sencillo de comprender para el usuario con poca experiencia y brinda algunas ventajas como:

- Evita la duplicidad de registros.
- Garantiza la integridad referencial.
- Facilita el proceso de normalización.

Es importante resaltar que el modelo relacional puede obtenerse de la conversión del **modelo entidad-relación**, eso sí, si el mismo fue bien planteado desde el principio. Cambiando la notación y aplicando ciertas reglas, el proceso de conversión no debería representar mucha dificultad.



De forma general, puede decirse que el **modelo E-R** es una técnica de diseño de bases de datos. Por su parte el **modelo relacional** es una formalización teórica de las bases de datos relacionales.

Los 3 aspectos más importantes al realizar la conversión:

- Atributos clave
- Cardinalidad
- Entidades fuertes y débiles

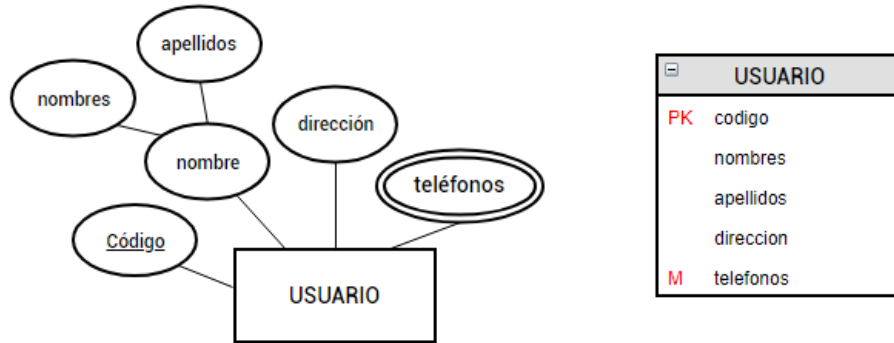
Conversión Entidad-Relación a Relacional

Una vez abstraída la información en el modelo E-R, proceder a su conversión al modelo relacional puede pensarse en un modelo en la que toda la información, de entidades y

relaciones, sea representada mediante **tablas**. Para esto, es necesario aplicar ciertas reglas de transformación. Se retomará el ejemplo de la biblioteca de la **Guía 1**.

Entidades Fuertes

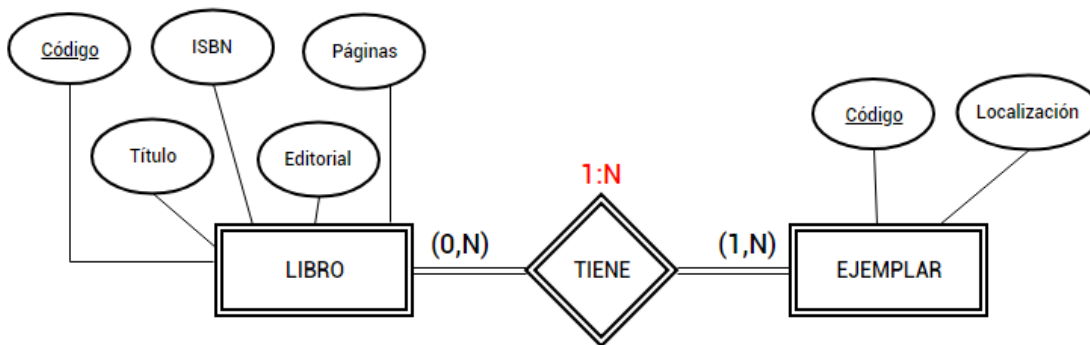
Las entidades fuertes son las que no dependen de ninguna otra para existir. Cada columna representará un atributo y cada **tupla** (fila) representará una entidad en particular.

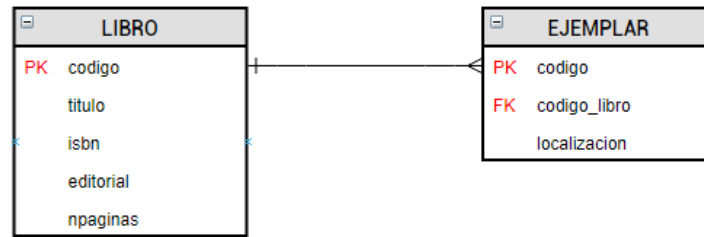


Los atributos se mantienen, incluyendo las claves primarias. Es recomendable renombrarlos para evitar ambigüedades con otras tablas que tengas atributos similares, (por ejemplo una tabla *Profesor* que también tienen nombre y teléfonos) aunque no es absolutamente necesario para estos casos. Las claves primarias van siempre al principio.

Conversión 1:N

Cuando se tiene una situación 1:N, la entidad del lado que tiene la **N** obtiene una copia de la **clave primaria** de la otra entidad como atributo de sí misma, como en el siguiente ejemplo.

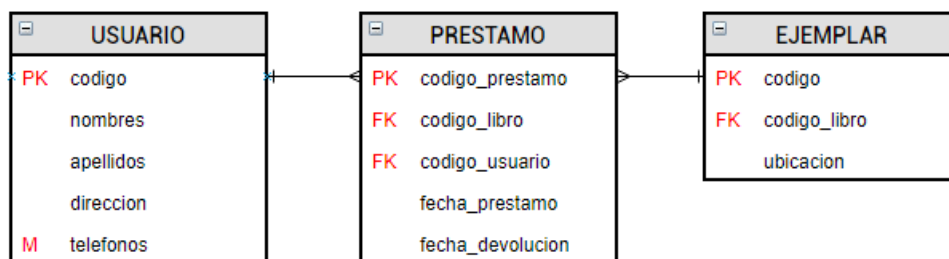
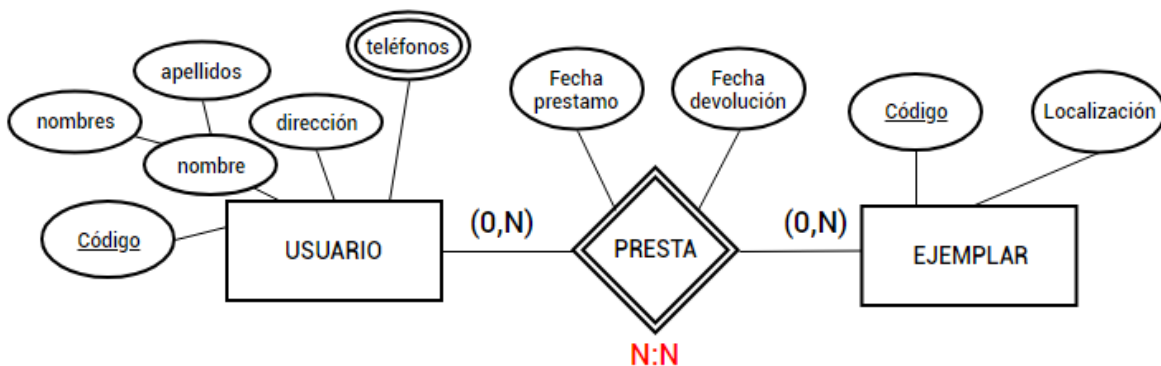




Como puede apreciarse, el **código del libro** pasa también como campo de la tabla Ejemplar, a este campo se le conoce como **llave o clave foránea**. La relación desaparece visualmente, pero la cardinalidad se mantiene. Tiene sentido que esto ocurra, ya que un ejemplar, es una copia de un libro en específico.

Las claves foráneas representan una relación directa con alguna entidad específica en otra tabla. En este caso, puede saberse de qué libro es cierto ejemplar observando el código de libro (FK).

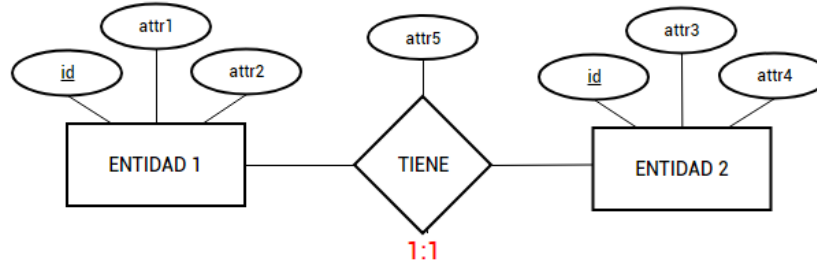
Conversión N:N



Para este caso, la relación se vuelve una tabla que tendrá su propia clave primaria y las claves primarias de las dos entidades involucradas, pero como claves foráneas. Además de estas, pueden agregarse atributos que se consideren necesarios. Para este caso se

añadieron los campos de **fecha de préstamo** y de **devolución**, ya que aportan información importante.

Conversión 1:1



Usualmente la más sencilla, aunque tiene algunos casos especiales.

| | |
|--|--|
| <p>Si la cardinalidad mínima de ambas entidades es 1, se obtiene una sola tabla que absorbe todos los atributos. Usualmente la tabla más significativa es la que absorbe.</p> | |
| <p>Si la cardinalidad mínima es 0 en uno de los lados, la entidad con 0 recibe como clave foránea la clave primaria de la otra entidad (similar al caso 1:N).</p> | |
| <p>Si la cardinalidad mínima en ambos lados de la relación es 0, se obtiene una nueva tabla, (similar al caso N:N).</p> | |

Aunque los últimos dos casos no son muy comunes, es importante conocerlos.

Normalización

Se dice que una base de datos está normalizada si está libre de ambigüedades e inconsistencias, es decir, permite tener un conjunto adecuado de relaciones.

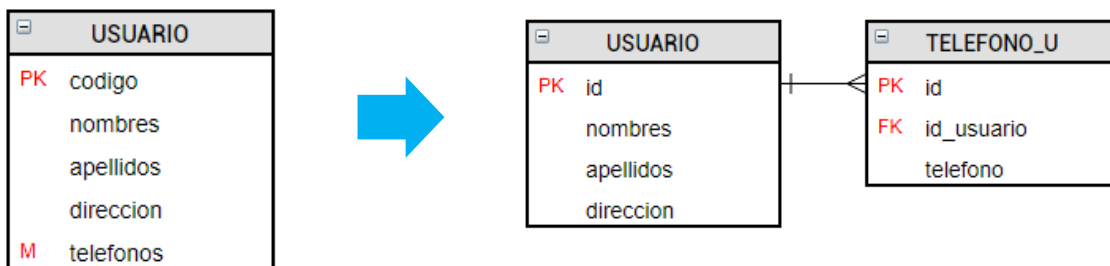
Si se ha diseñado correctamente el diagrama de la base desde el principio, el proceso de normalización se facilita en gran medida. Este proceso se lleva a cabo aplicando una serie de reglas conocidas como **formas normales**.



Se dice que una base de datos está en su “N forma normal” si **todas** sus tablas cumplen con las reglas de esa N forma normal.

Primera forma normal

La principal característica de esta forma se resume en que *“cada intersección de fila y columna contiene exactamente un valor del dominio aplicable (y nada más)”*. Es decir, no se permite almacenar **grupos de valores** para un solo registro. Usualmente esto ocurre cuando se han definido atributos multivaluados.



Como puede observarse, de esta forma pueden almacenarse cualquier cantidad de teléfonos para un usuario, sin necesidad de guardarlo directamente como un grupo.

Segunda forma normal

Como requisito, las tablas deben cumplir con la primera forma normal. Esta forma dice que todos los atributos que no son parte de la llave primaria, deben depender de toda la llave primaria y no solo parte de ella.

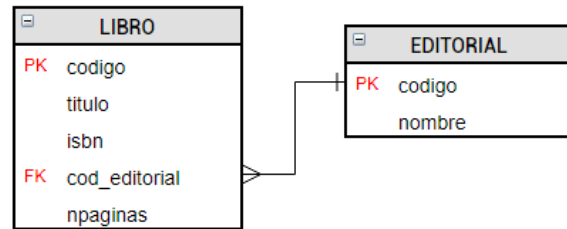
Visto de otra forma, si un atributo (no primario) está siendo redundante en los datos que almacena (se aplican a varios registros) probablemente está queriendo actuar de forma independiente y se deba formar una nueva tabla. Las tablas resultantes de esta forma suelen llamarse tablas **catálogo**.

Retomando el ejemplo para explicar esta situación:

| LIBRO | | | | |
|--------|----------------------|-----------|----------------|-------------------|
| Código | Título | ISBN | Editorial | Número de páginas |
| 0001 | La Casa Torcida | 978607074 | Booket | 240 |
| 0002 | La línea | 234234234 | Paidos | 260 |
| 0003 | Poirot investiga | 234920438 | Booket | 260 |
| 0004 | Animales Fantásticos | 876801832 | Harper Collins | 130 |

Como puede apreciarse en el ejemplo, si se necesita registrar más libros de la editorial *Booket* la tabla empezará a tener datos repetidos, al igual que con otras editoriales. Visto de otro modo, este campo puede actuar de forma independiente como una tabla.

Separando la editorial como una tabla y relacionando esta nueva tabla con una clave foránea, se resuelve el problema de redundancia.



| LIBRO | | | | |
|--------|----------------------|-----------|---------------------|------------|
| Código | Título | ISBN | Código de editorial | N. páginas |
| 0001 | La Casa Torcida | 978607074 | 001 | 240 |
| 0002 | La línea | 234234234 | 002 | 260 |
| 0003 | Poirot investiga | 234920438 | 001 | 260 |
| 0004 | Animales Fantásticos | 876801832 | 003 | 130 |

| EDITORIAL | |
|-----------|----------------|
| Código | Nombre |
| 001 | Booket |
| 002 | Paidos |
| 003 | Harper Collins |

Tercera Forma Normal

Una tabla está en tercera forma normal si no existe ninguna dependencia transitiva entre los atributos que no son clave. Además debe cumplir también con la 2FN. Un ejemplo sencillo se da en el siguiente caso, donde se tiene una tabla de eventos y el lugar donde se realizará cada uno.

| EVENTO | | | |
|--------|--------------------------|-------------------|-------------------|
| Código | Nombre | Local | Dirección |
| 0001 | Metallica | Estadio Mágico G. | 49 Av. Sur |
| 0002 | Tributo Soda Stereo | Teatro Presidente | C. Circunvalación |
| 0003 | Día Nacional de la Salsa | CIFCO | Av. La Revolución |
| 0004 | Adele Sinfónico | Teatro Presidente | C. Circunvalación |

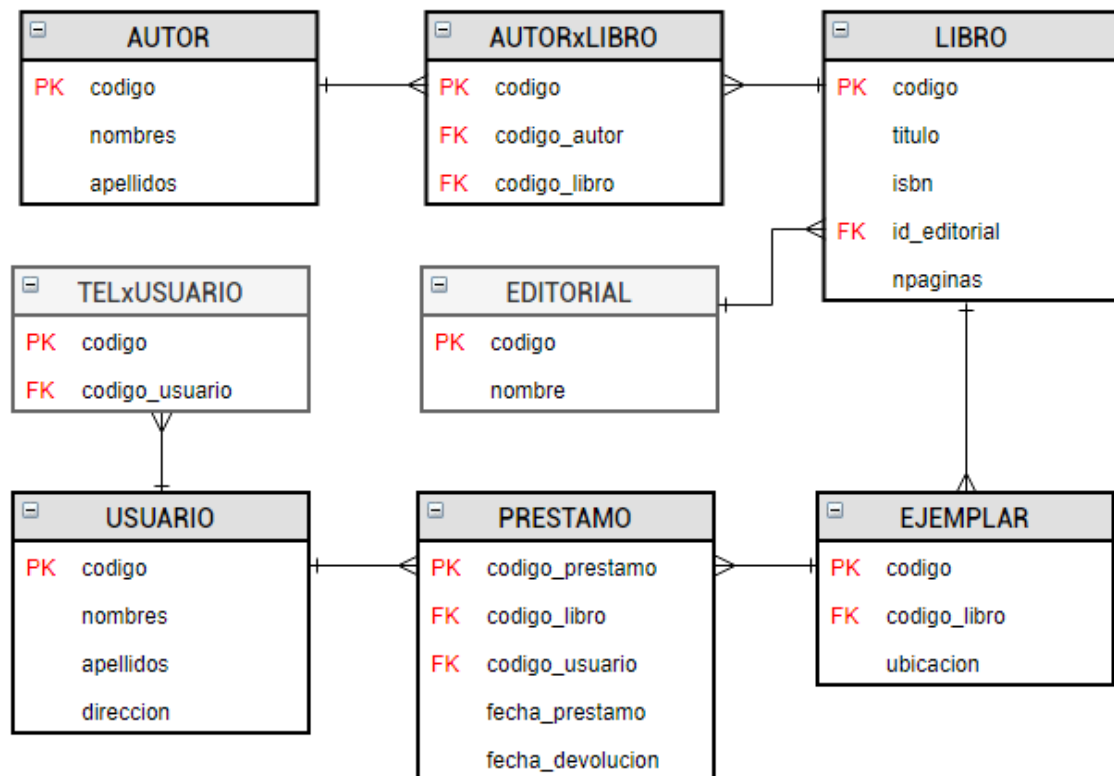
La 3F falla debido a que la dirección de local depende directamente del local y éste último no es un atributo clave. Esto puede llevar a problemas de actualización donde nada impide tener diferentes direcciones para el mismo local. Esto se resuelve fácilmente creando una nueva tabla que contenga este registro dependiente.

| EVENTO | | |
|--------|--------------------------|-----------------|
| Código | Nombre | Código de Local |
| 0001 | Metallica | 001 |
| 0002 | Tributo Soda Stereo | 002 |
| 0003 | Día Nacional de la Salsa | 003 |
| 0004 | Adele Sinfónico | 002 |

| LOCAL | | |
|--------|-------------------|-------------------|
| Código | Nombre | Dirección |
| 001 | Estadio Mágico G. | 49 Av. Sur |
| 002 | Teatro Presidente | C. Circunvalación |
| 003 | CIFCO | Av. La Revolución |

La idea final de la 3FN es separar los grupos de datos que tengan dependencias transitivas a tablas separadas e identificar cada grupo con una llave, tal como se ha logrado en el ejemplo anterior.

El ejemplo de la librería, ya normalizada, quedaría de la siguiente manera.



Prestar atención a la dirección de las relaciones obtenidas de la 1FN y 2FN

Ejercicio

Tomando como base el diagrama E-R obtenido del ejercicio realizado de la primera guía (sea el de la compañía aérea o de la tienda de electrónicos), realizar su conversión al modelo relacional y luego aplicar las 3 formas normales (dos diagramas).