

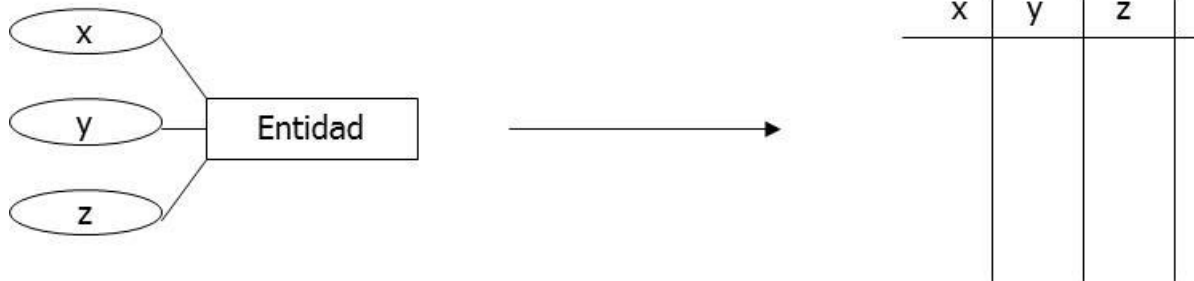
MODELO RELACIONAL

Sistemas de bases de datos I

Universidad de Caldas

Introducción

En este modelo la base de datos es vista por el usuario como una relación de tablas. Cada fila de la tabla es un registro o tupla y los atributos con columnas o campos.



Elementos básicos

- RELACIÓN

Es la estructura básica del modelo relacional. Se representa mediante una tabla.

- ATRIBUTO

Representa las propiedades de la relación. Se representa mediante una columna.

- DOMINIO

Es el conjunto válido de valores que toma un atributo.

- TUPLA

Es una ocurrencia de la relación. Se representa mediante una fila.

- GRADO DE UNA RELACIÓN= Numero de atributos o columnas

- CARDINALIDAD= Numero de filas o tuplas de una relación

Elementos básicos

- La relación es el elemento fundamental del modelo relacional (de ahí el nombre del modelo), y se puede representar en forma de tabla:

Conceptos

- **Relación:** Película (título, año, duración)

Tuplas

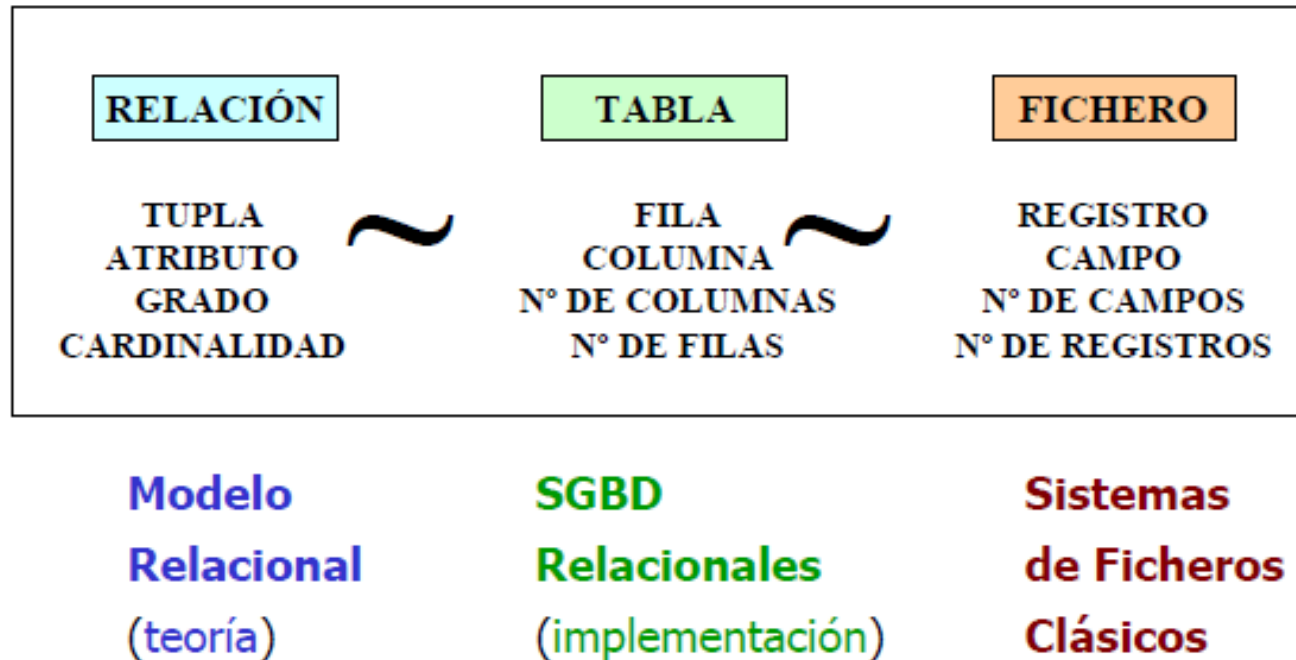
Título	Año	Duración
La guerra de las galaxias	1977	123
El señor de los anillos I	2001	178
Mar adentro	2004	125

Dominio=textos **Dominio**=enteros

Cardinalidad=3

Grado de la relación=3

Terminología



Claves o llaves

- Clave Candidata (Candidate Key): conjunto de atributos que identifican unívoca y mínimamente cada tupla de la relación.
- De la definición de relación se deriva que siempre existe, al menos, una clave candidata (al ser una relación un conjunto y no existir dos tuplas iguales, el conjunto de todos los atributos siempre tiene que identificar unívocamente a cada tupla).
- Una relación puede tener más de una clave candidata. En este caso se debe distinguir entre:
 - Clave Primaria (Primary Key):

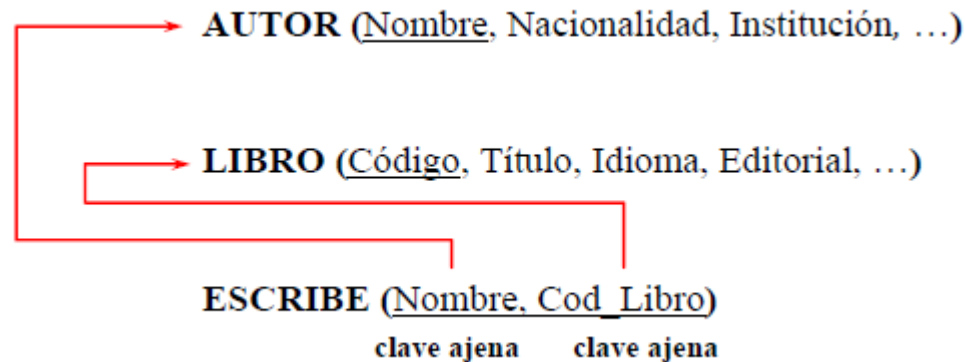
Es la clave candidata que el usuario escoge para identificar las tuplas de la relación.

Cuando sólo existe una clave candidata, ésta es la clave primaria.
 - Claves Alternativas (Alternative Key):

Las claves candidatas que no han sido escogidas como clave primaria.

Clave ajena o llave foránea

- Atributo de una relación que es clave primaria en otra relación.



- Los atributos principales (forman la clave primaria) se subrayan las claves ajenas se indican con una flecha

Restricciones de Integridad

- Integridad de la Entidad: Ningún componente de la clave primaria puede tomar valores nulos o desconocidos, porque entonces no se podrían distinguir dos entidades.
- Integridad Referencial: Cualquier valor que tome un atributo en una relación del que es clave foránea, debe existir en la relación del que es clave primaria.

Restricciones de integridad semánticas

- Son definidas por el usuario.
- Son facilidades que el modelo ofrece a los usuarios/diseñadores para que puedan reflejar en el esquema, lo más fielmente posible, la semántica del mundo real.
- Los tipos de restricciones semánticas permitidos en el MR (incorporados a SQL 92) son:
 - Clave Primaria (PRIMARY KEY),
 - Unicidad (UNIQUE),
 - Obligatoriedad (NOT NULL),
 - Integridad Referencial (FOREIGN KEY),
 - Restricciones de Rechazo:
 - Verificación (CHECK), y
 - Aserción (CREATE ASSERTION).
- Disparador (trigger), incluido en SQL3 pero no en SQL92.
- Dependencia (se estudiarán más tarde).

Conversión del modelo conceptual al modelo relacional:

- Conversión de Entidades:
- Cada entidad de diagrama Entidad/Relación se transforma directamente en una tabla. Los atributos de la entidad pasan a ser automáticamente las columnas de la tabla.

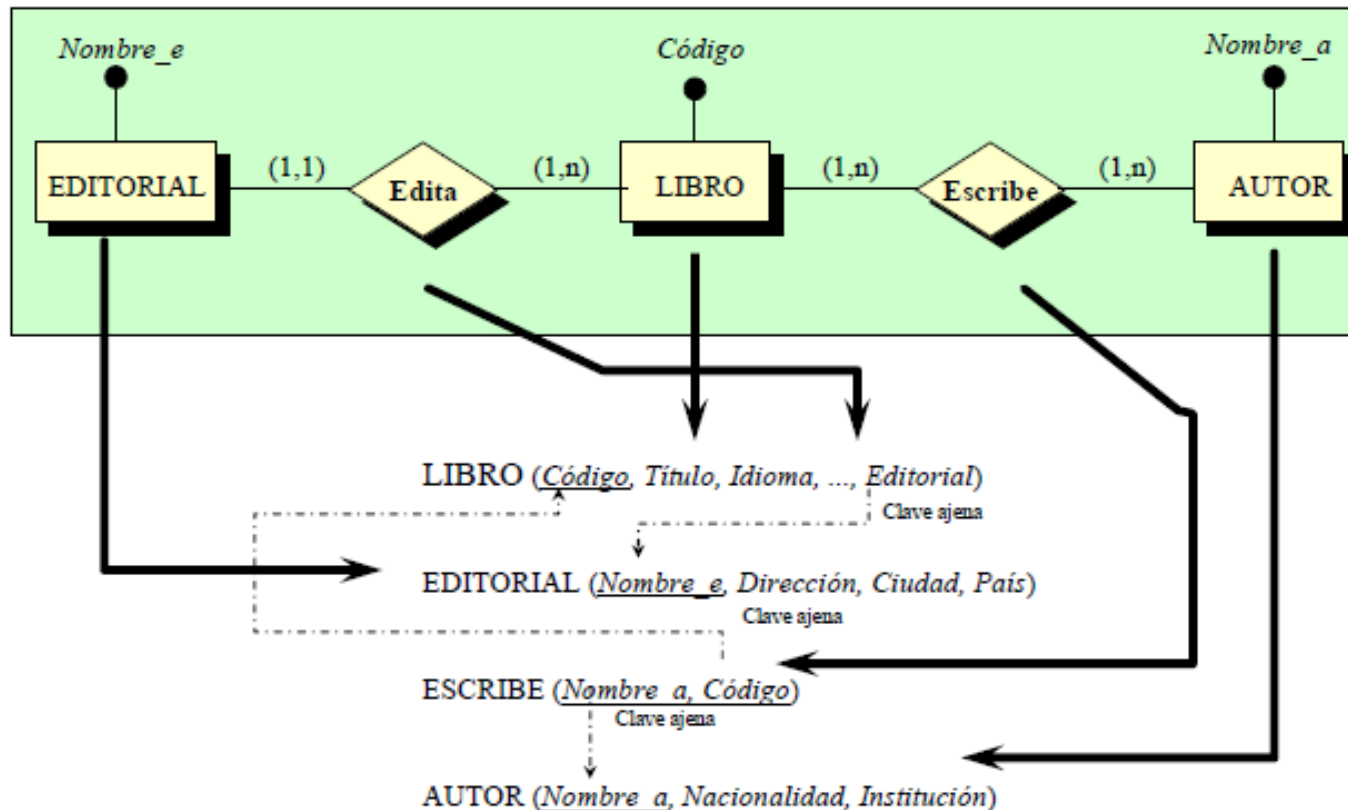
Entidad —————> Tabla

Atributos —————> Columnas

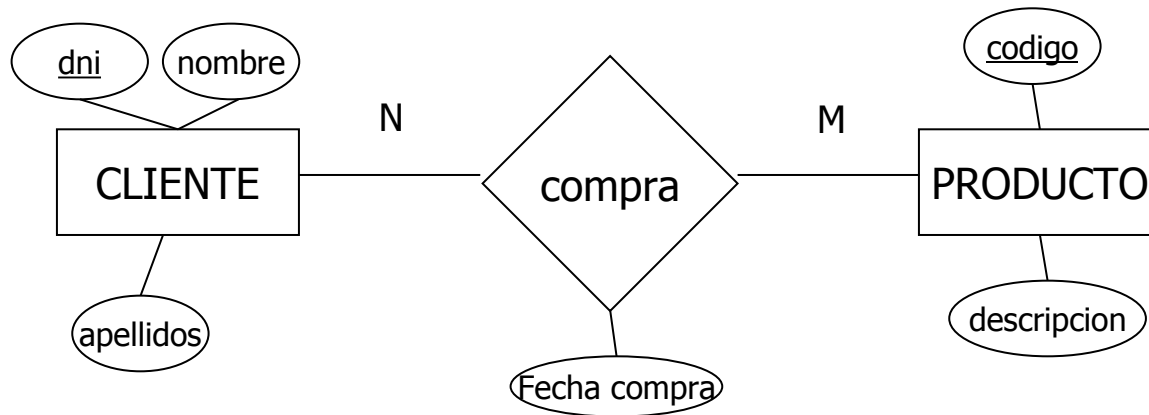
Diseño Lógico Estándar

- Las **tres reglas básicas** para convertir un esquema en el modelo ER al relacional son las siguientes:
 1. Todo tipo de entidad se convierte en una relación.
 2. Todo tipo de interrelación N:M se transforma en una relación.
 3. Para todo tipo de interrelación 1:N se realiza lo que se denomina **propagación de clave** (regla general), o se crea una nueva relación.
- Como el modelo relacional no distingue entre entidades y relaciones, ambos conceptos deben representarse mediante relaciones. Esto implica una **pérdida de semántica** con respecto al esquema ER:
- Las **relaciones N:M no se distinguen de las entidades** (ambas se transforman en tablas)
- Las **relaciones 1:N** se suelen representar mediante una **propagación de clave**, desapareciendo incluso el nombre de la interrelación.

Ejemplo de transformación

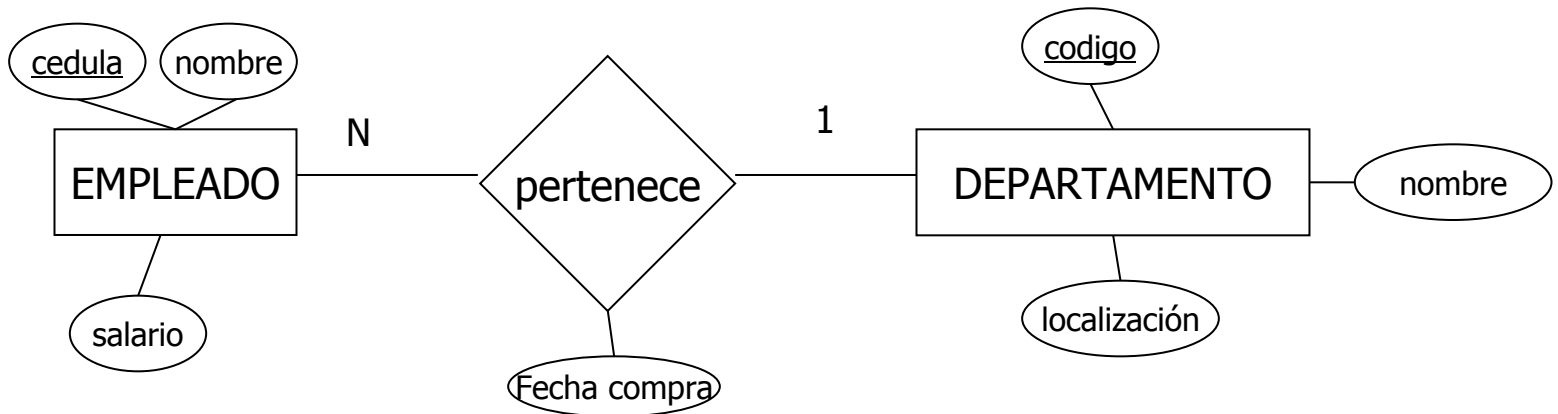


Ejemplo: Relaciones N:M



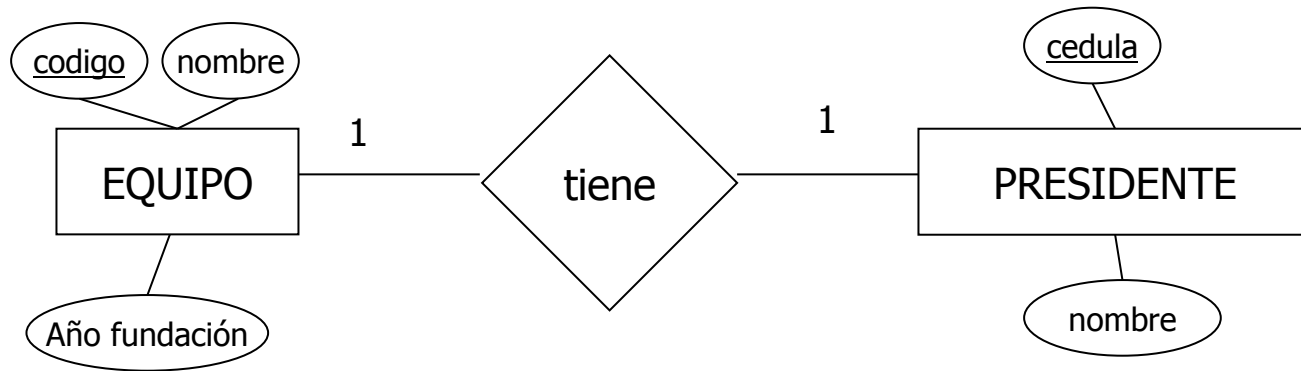
- CLIENTE (**dni**, nombre, apellidos)
- PRODUCTO (**código**, descripción)
- COMPRAS (**dni_cliente**, **código _ producto**, fecha_compra)

Ejemplo: Relaciones 1:N



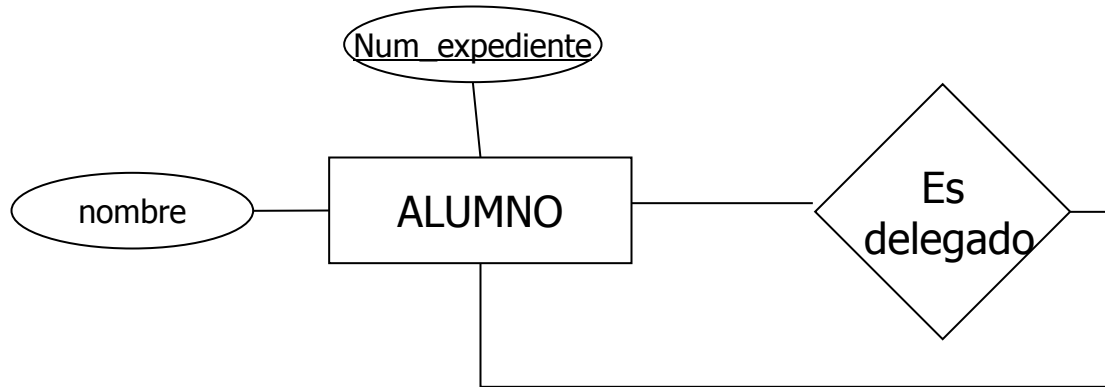
- EMPLEADO (**cedula**, nombre, salario, codigo_departamento)
- DEPARTAMENTO (**codigo**, nombre, localización)

Ejemplo: Relaciones 1:1



- EQUIPO (**codigo**, nombre, año _ fundación)
- PRESIDENTE (**cedula**, nombre, codigo _ equipo)
- EQUIPO (**codigo**, nombre, año _ fundación, ced_presidente)
- PRESIDENTE (**cedula**, nombre)

Ejemplo: Relaciones reflexivas



- ALUMNO (**num_expediente**, nombre, num_expediente_delegado)

Dominios y atributos

- **Transformación de Atributos de Entidades:**

Cada atributo de una entidad se transforma en una columna de la tabla a la que ha dado lugar la entidad. Teniendo en cuenta los identificadores, esta regla se divide en tres subreglas:

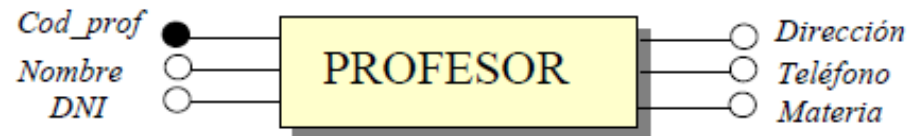
- **Atributos Identificadores:** Los atributos que son identificadores principales (AIP) pasan a formar la clave primaria de la tabla.

En SQL se representan con la cláusula **PRIMARY KEY** dentro de la orden **CREATE TABLE**.

- **Atributos Identificadores Alternativos:** Se representan en SQL por medio de la cláusula **UNIQUE** dentro de la orden **CREATE TABLE**.
- **Atributos No Identificadores:** Se representan como columnas de la tabla correspondiente.

Dominios y atributos, ejemplo

EER



MR-SQL

```
CREATE TABLE Profesor (  
    Cod_Profesor Códigos,  
    Nombre Nombres,  
    DNI DNIS, NOT NULL  
    Dirección Lugares,  
    Teléfono Nos_Teléfono,  
    Materia Materias,  
    PRIMARY KEY (Cod_Profesor),  
    UNIQUE (DNI));
```

- Ejem
atributos

us

Transformación de relaciones N:M

- **Transformación de relaciones N:M.**
- Un tipo de **relación N:M** se transforma en una **tabla** que tendrá como clave primaria la concatenación de los AIP de los tipos de entidad que asocia. Además, cada uno de los atributos que forman la clava primaria de esta tabla también son claves ajenas que referencian a las tablas en que se han convertido las entidades relacionadas (claves primarias):
- En **SQL** se representan con la cláusula **FOREIGN KEY** dentro de la sentencia de creación de la tabla.
- Para cada clave ajena así obtenida deberá estudiarse cuales son los **modos de borrado y modificación** adecuados (opciones **ON DELETE** y **ON UPDATE** en SQL). Las opciones permitidas en SQL son:
- Operación restringida (en caso de no especificar la acción o poner **NO ACTION**)
- Puesta a nulo (**SET NULL**)
- Puesta a valor por defecto (**SET DEFAULT**)
- Operación en cascada (**CASCADE**).
- Las **cardinalidades mínimas** de las entidades participantes en la relación se pueden modelar utilizando restricciones **CREATE ASSERTION**.

Transformación de relaciones 1:N

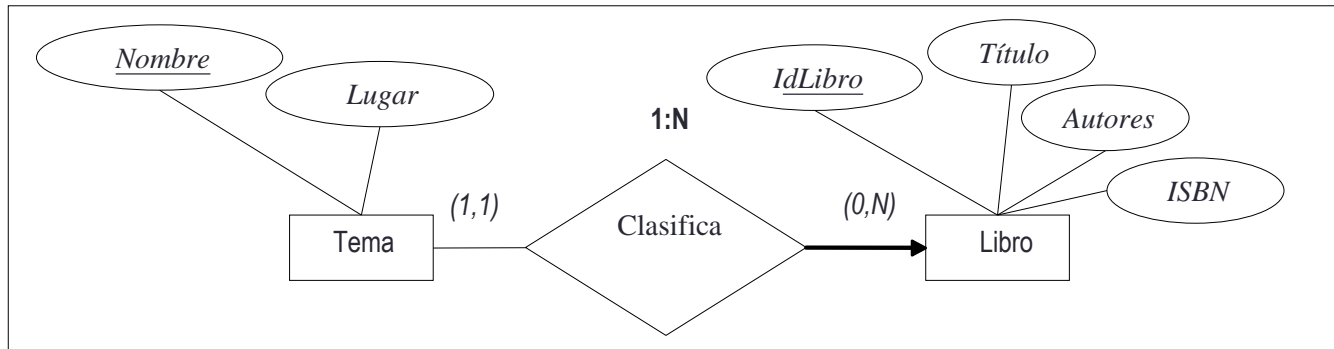
- **Transformación de relaciones 1:N.** Existen **dos soluciones**:
 - a) Propagar los AIP del tipo de entidad que tiene de cardinalidad máxima 1 a la que tiene N (**propagación de clave**). Esta es la regla habitual.
 - b) Transformar la relación en una **tabla** como si se tratara de una interrelación N:M; pero ahora la clave primaria de la tabla creada es sólo la clave primaria de la tabla a la que le corresponde la cardinalidad máxima n.

La opción b) se utiliza sólo cuando:

- El número de ejemplares relacionados de la entidad que propaga su clave es muy pequeño y, por tanto, existirían muchos valores nulos en la clave propagada.
- Se prevé que la relación acabará convirtiéndose en una N:M.
- La relación tiene atributos propios y no es deseable propagarlos (a fin de conservar la semántica).

Ejercicios práctico 1

- **Clasificación de libros y temas en una biblioteca:** De cada tema puede haber varios libros o ninguno. Cada libro obligatoriamente está registrado en algún tema.



Ejercicio práctico 2

- **Un Videoclub (clientes y películas):**

- Supongamos que tenemos un videoclub, y lo queremos informatizar. Los socios del videoclub alquilan las películas, por tanto se trata de una relación binaria (de grado 2) entre la entidad socio y la entidad película. Un socio puede alquilar muchas películas cada vez, pero cada película sólo puede ser alquilada por un socio cada vez. Por tanto la relación es del tipo 1:N, este factor determina el número de entidades asociadas dentro de la relación.
- La cardinalidad de una entidad identifica el número de ocurrencias máximo y mínimo que participan en una relación. La cardinalidad de la entidad Socio es (1,1) pues en un alquiler hay uno y sólo un socio, mientras que la cardinalidad de la entidad Película es (1,N) pues en un alquiler hay al menos una película y podrías ser más. En cada operación de alquiler debe poder registrarse la fecha de alquiler, y la fecha de devolución de la película alquilada.

