

TEMA 2.- TRANSFORMACIÓN DEL MODELO LÓGICO EN RELACIONAL

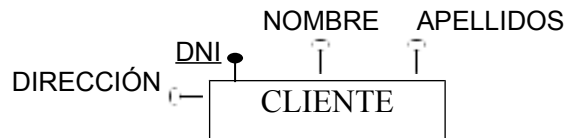
Índice

1. Transformación del esquema conceptual al esquema lógico basado en el modelo relacional.
 - 1.1. Transformación de tipos de entidad y de sus atributos.
 - 1.2. Transformación de tipos de interrelación N:M.
 - 1.3. Transformación de tipos de interrelación 1:N.
 - 1.4. Transformación de tipos de interrelación 1:1.
 - 1.5. Transformación de interrelaciones débiles.
 - 1.6. Transformación de tipos de interrelación de grado 3.
 - 1.7. Transformación de supertipos y subtipos.
2. Restricciones del modelo relacional.
 - 2.1. Características propias de una relación.
 - 2.2. Regla de integridad de entidad.
 - 2.3. Regla de integridad referencial.
 - 2.4. Restricciones semánticas o de usuario

1. TRANSFORMACIÓN DEL ESQUEMA CONCEPTUAL AL ESQUEMA LÓGICO BASADO EN EL MODELO RELACIONAL.

1.1. TRANSFORMACIÓN DE TIPOS DE ENTIDAD Y DE SUS ATRIBUTOS.

Toda entidad se transforma en tabla en el esquema relacional manteniendo el número y tipo de los atributos. El atributo de clave primaria pasa a ser la clave primaria de la tabla y el resto de atributos pasan a ser columnas de la tabla. La clave primaria se subraya en la tabla. Ejemplo:



CLIENTE (dni, nombre, apellidos, dirección).

DNI	NOMBRE	APELLIDOS	DIRECCIÓN
1	PEPE
2	RICARDO
3	JUAN

Hay que indicar también los atributos derivados:

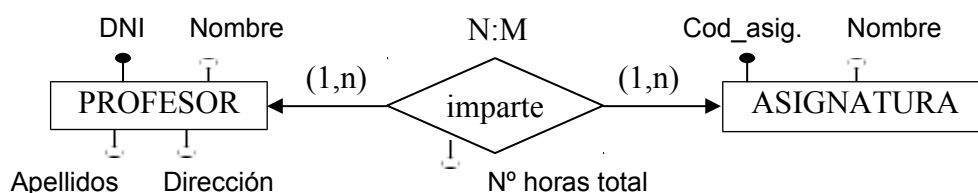
PARTIDO(Cod_partido, Nombre, Votos_obtenidos D)

- Cada atributo de la entidad se transforma en una columna en la relación a la que ha dado lugar la entidad. Veamos como se definen cada uno de los tipos de atributos:
 - El o los atributos principales de una entidad (clave primaria) pasan a ser la clave primaria de la relación. No pueden ser nulos.
 - El resto de los atributos pasan a ser columnas de la tabla, pudiendo tomar valores nulos, a no ser que se indique lo contrario por restricciones de nuestro sistema de información.

1.2. TRANSFORMACIÓN DE TIPOS DE INTERRELACIÓN N:M.

Se crea una nueva tabla que incluye los atributos de la propia relación (si los tuviera) y las claves primarias de las dos entidades, que forman la clave primaria de la nueva relación.

Ejemplo:



PROFESOR (DNI, Nombre, Apellidos, Dirección).

ASIGNATURA (Cod_asig, Nombre).

IMPORTE (DNI, Cod_asig, Nº horas total).

FK1 (Cod_asig) / ASIGNATURA.

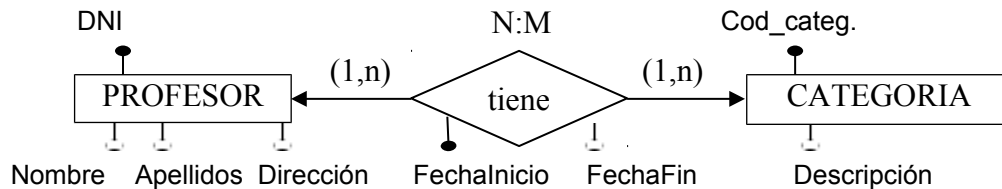
-- No tenemos que indicar NOT NULL porque

FK2 (DNI) / PROFESOR.

-- son atributos de la clave primaria

Cada uno de los atributos que forman la clave primaria de esta tabla son clave ajena respecto a cada una de las tablas donde este atributo es clave primaria, lo que se indica con **FOREIGN KEY**.

Ejemplo:



PROFESOR (DNI, Nombre, Apellidos, Dirección).

CATEGORIA (Cod_categ., Descripción).

TIENE (DNI, Cod_Categ., FechaInicio, FechaFin).

FK1 (DNI) / PROFESOR.

-- No tenemos que indicar NOT NULL porque

FK2 (Cod_categ.) / CATEGORIA.

-- son atributos de la clave primaria

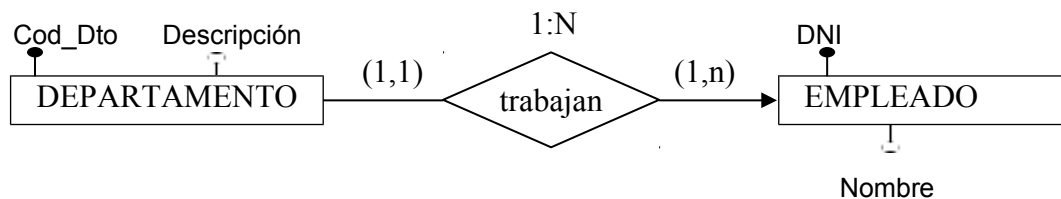
1.3. TRANSFORMACIÓN DE TIPOS DE INTERRELACIÓN 1:N.

Existen 2 casos:

1. Cardinalidad (1,1)

Propagar la clave principal de la entidad que tiene cardinalidad máxima 1 a la que tiene N, y hacer desaparecer la tabla de la relación como tal.

Ejemplo:



DEPARTAMENTO (Cod_Dto, Descripción).

EMPLEADO (DNI, Nombre, Cod_Dto).

FK (Cod_Dto) / DEPARTAMENTO NOT NULL.

2. Cardinalidad (0,1)

Se puede hacer como en la cardinalidad (1,1) ó se puede crear una nueva tabla con la clave de la entidad del lado N como si se tratase de una relación N:M. ¿De qué depende? Hay que tener en cuenta que si se hace como las (1,1), al ser el mínimo 0, puede haber claves ajenas nulas. Entonces, si se prevé que esto va a ocurrir en gran número de veces, es decir, que vamos a tener muchas ocurrencias con la clave ajena a nula, entonces es mejor crear una nueva tabla.

1.4. TRANSFORMACIÓN DE TIPOS DE INTERRELACIÓN 1:1.

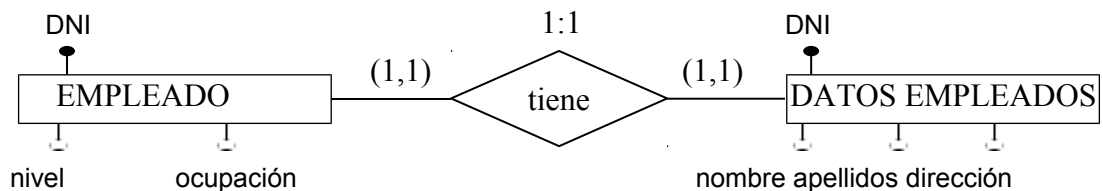
Tenemos 3 Casos {

- Cardinalidades (1,1) y (1,1).
- Cardinalidades (1,1) y (0,1).
- Cardinalidades (0,1) y (0,1).

1. Cardinalidades (1,1) y (1,1).

- Si los dos tipos de entidad tienen el mismo identificador: los dos tipos de entidad se transforman en una única tabla formada por la agregación de los atributos de los dos tipos de entidad. La clave de la tabla es el identificador de los tipos de entidad (es el mismo en ambos). Si no interesase esta solución, se pueden dejar dos tablas separadas y no habría propagación ya que tienen el mismo identificador.

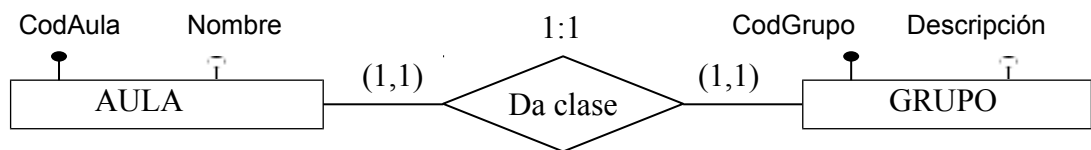
Ejemplo:



EMPLEADO (DNI, nivel, ocupación, nombre, apellidos, dirección).

- Si los dos tipos de entidad tienen diferente identificador: cada tipo de entidad se transforma en una tabla y se propaga la clave de cualquiera de ellas a la tabla resultante de la otra.

Ejemplo:



1)

AULA (CodAula, Nombre).
GRUPO (CodGrupo, Descripción, CodAula).
FK (CodAula) / AULA NOT NULL

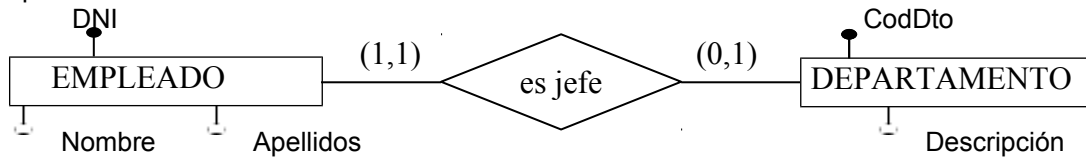
2)

AULA (CodAula, Nombre, CodGrupo)
FK (CodGrupo) / GRUPO NOT NULL
GRUPO (CodGrupo, Descripción).

2. Cardinalidades (1,1) y (0,1).

Se propaga la clave de la entidad con cardinalidad (1,1) a la tabla resultante de la entidad con cardinalidad (0,1).

Ejemplo:

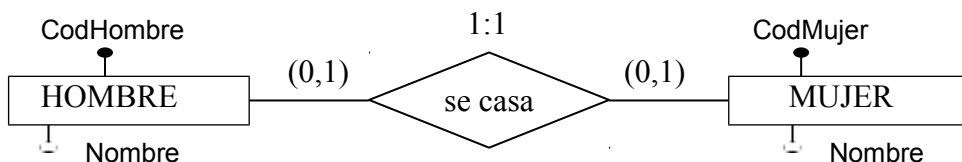


EMPLEADO (DNI, Nombre, Apellidos).
 DEPARTAMENTO (CodDep, Descripción, DNI_jefe).
 FK (DNI_jefe) / EMPLEADO NOT NULL.

3. Cardinalidades (0,1) y (0,1).

Se propaga una de las dos claves en el sentido que se considere más apropiado o bien se crea una nueva tabla. La segunda opción es mejor para evitar tener muchos nulos como propagación de alguna de las claves a la otra.

Ejemplo:



1)

HOMBRE (CodHombre, Nombre).
 MUJER (CodMujer, Nombre, CodHombre).
 FK (CodHombre) / HOMBRE.
 CodHombre UNIQUE.

2)

HOMBRE (CodHombre, Nombre, CodMujer).
 FK (CodMujer) / MUJER.
 CodMujer UNIQUE.
 MUJER (CodMujer, Nombre).

3)

HOMBRE (CodHombre, Nombre).
 MUJER (CodMujer, Nombre).
 SE CASA (CodHombre, CodMujer).
 FK1 (CodHombre) / HOMBRE.
 FK2 (CodMujer) / MUJER.
 CodMujer UNIQUE
 CodHombre UNIQUE

Ó

CodHombre, CodMujer
 FK1 (CodHombre) / HOMBRE.
 FK2 (CodMujer) / MUJER NOT NULL.
 CodMujer UNIQUE

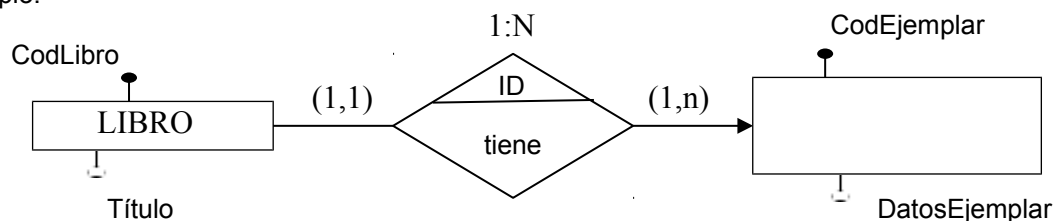
1.5. TRANSFORMACIÓN DE INTERRELACIONES DÉBILES.

La debilidad en existencia se trata como una interrelación normal.

En la debilidad en identificación la entidad débil se transforma en una tabla, propagando la clave de la entidad fuerte, que pasa a formar parte de la clave primaria de la entidad débil.

En la semántica no recogida hay que indicar borrado en cascada de la débil cuando se borre la fuerte.

Ejemplo:

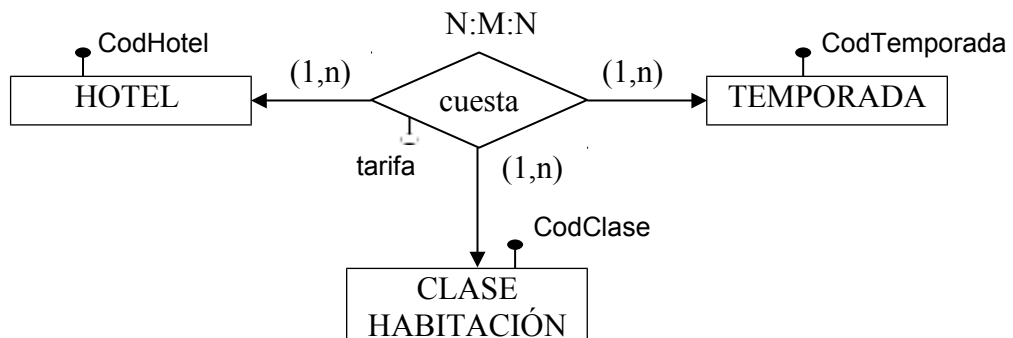


LIBRO (CodLibro, Título).
 EJEMPLAR (CodEjemplar, CodLibro, DatosEjemplar).
 FK (CodLibro) / LIBRO.

1.6. TRANSFORMACIÓN DE TIPOS DE INTERRELACIÓN DE GRADO 3.

La interrelación se transforma en una tabla cuya clave primaria será la concatenación de las claves primarias de las tres entidades que relaciona. Los atributos de la interrelación pasan a ser también columnas de la tabla. Cada uno de los atributos que forman parte de la clave primaria son a su vez claves ajenas respecto a cada una de las tablas donde dicho atributo es clave primaria.

Ejemplo:



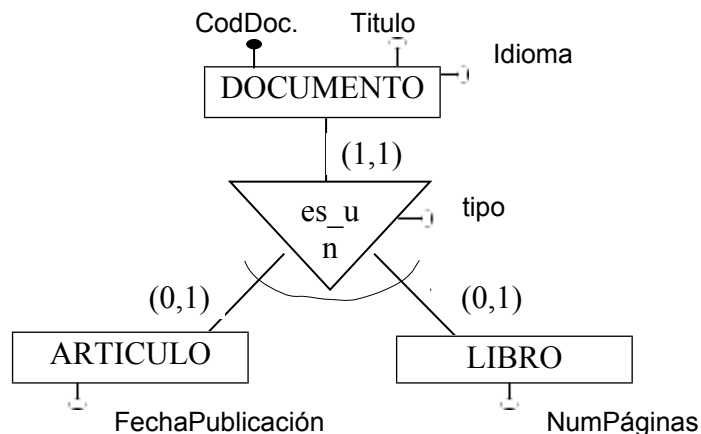
HOTEL (CodHotel,).
 TEMPORADA (CodTemporada,).
 CLASEHABITACIÓN (CodClase,).
 CUESTA (CodHotel, CodTemporada, Codclase, tarifa).
 FK1 (CodHotel) / HOTEL.
 FK2 (CodTemporada) / TEMPORADA.
 FK3 (CodClase) / CLASEHABITACIÓN.

Para completar este apartado, consultar el documento **Ampliación de Relaciones Ternarias**.

1.7. TRANSFORMACIÓN DE SUPERTIPOS Y SUBTIPOS.

Tanto el supertipo como los subtipos se transforman en tablas. Los subtipos tendrán como clave primaria el atributo de clave primaria del supertipo y sus atributos pasarán a ser columnas de la tabla. El supertipo se transforma como una entidad y se le añade el atributo discriminante como columna de la tabla.

Ejemplo:



Parcial/Exclusiva (0,1)

1ª Forma

ARTICULO(CodDoc, FechaPublicación).
FK (CodDoc) / DOCUMENTO.
LIBRO (CodDoc, NumPáginas).
FK (CodDoc) / DOCUMENTO.
DOCUMENTO (CodDoc, Titulo, Idioma, tipo).
tipo check('a','l','o'). ó también puede 'a', 'l' o ser **nulo** para
identificar el tercer tipo.

Nota: el atributo tipo puede tener el dominio que se crea mejor, por ejemplo
también podrían haberse considerado los valores 1,2 o 3.

2ª Forma

ARTICULO (CodDoc, FechaPublicación).
FK (CodDoc) / DOCUMENTO.
LIBRO (CodDoc, NumPáginas).
FK (CodDoc) / DOCUMENTO.
DOCUMENTO (CodDoc, Titulo, Idioma).
ES_TIPO (CodDoc, tipo).
tipo check('a','l').

Total/Exclusiva (1,1)

ARTICULO(CodDoc, FechaPublicación).
FK (CodDoc) / DOCUMENTO.
LIBRO (CodDoc, NumPáginas).
FK (CodDoc) / DOCUMENTO.
DOCUMENTO (CodDoc, Titulo, Idioma, tipo).
tipo check('a','l').

Parcial/Inclusiva (0,N)

ARTICULO (CodDoc, FechaPublicación).
FK (CodDoc) / DOCUMENTO.
LIBRO (CodDoc, NumPáginas).
FK (CodDoc) / DOCUMENTO.
DOCUMENTO (CodDoc, Titulo, Idioma).
ES_TIPO (CodDoc, tipo).
tipo check('a','l').

En la tabla DOCUMENTO pueden haber algunos de ellos que no estén ni en
ARTICULO ni en LIBRO.

Total/Inclusiva (1,N)

ARTICULO (CodDoc, FechaPublicación).
FK (CodDoc) / DOCUMENTO.
LIBRO (CodDoc, NumPáginas).
FK (CodDoc) / DOCUMENTO.
DOCUMENTO (CodDoc, Titulo, Idioma).
ES_TIPO (CodDoc, tipo).
tipo check('a','l').

El sistema debe asegurar que todos los documentos que estén en DOCUMENTO
deben estar o en ARTICULO o en LIBRO o en ambas tablas.

2. RESTRICCIONES DEL MODELO RELACIONAL.

La relación es el elemento básico del modelo relacional y se puede representar como una tabla. Todos los datos de una base de datos se representan en forma de relaciones cuyo contenido varía en el tiempo.

2.1. CARACTERÍSTICAS PROPIAS DE UNA RELACIÓN.

Dichas características se han de cumplir obligatoriamente y son las siguientes:

- No hay dos tuplas iguales.

Ejemplo:

	<u>DNI</u>	Nombre	Apellidos
Tupla -----	1	Pepe	García
	2	Juan	González

- El orden de las tuplas no es significativo.

Ejemplo:

	<u>DNI</u>	Nombre	Apellidos
Tupla -----	2	Juan	González
	1	Pepe	García

- El orden de los atributos no es significativo.

Ejemplo:

<u>DNI</u>	Apellidos	Nombre	<u>DNI</u>	Nombre	Apellidos
1	García	Pepe	1	Pepe	García
2	González	Juan	2	Juan	González

- Cada atributo sólo puede tomar un único valor del dominio, no admitiéndose por tanto los grupos repetitivos.

2.2. REGLA DE INTEGRIDAD DE ENTIDAD.

Ningún atributo que forme parte de la clave primaria de una relación puede tomar un valor nulo, es decir, un valor desconocido o inexistente. Las claves ajenas sí pueden tomar valores nulos.

Ejemplo:

<u>DNI</u>	Nombre
1	Pepe
2	Juan
-	Luis

2.3. REGLA DE INTEGRIDAD REFERENCIAL.

Si una relación R2 tiene una clave ajena de una relación R1, todo valor de dicha clave ajena debe concordar con un valor de la clave primaria de R1 o ser **nulo**.

Casos Prácticos

Ejemplo1: RELACIÓN COCHES.

<u>MODELO</u>	COLOR	<u>MODELO</u>	COLOR
Renault Megane	Verde	Seat Ibiza	Blanco
Ford Focus	Azul	Renault Megane	Verde
Seat Ibiza	Blanco	Ford Focus	Azul

La relación coches ¿es la misma en ambos casos?.

Sí es la misma relación, ya que el orden de las tuplas no es significativo.

Ejemplo2: RELACIÓN COCHES. Y ahora, ¿es la misma relación?

<u>MODELO</u>	COLOR	PRECIO	<u>MODELO</u>	PRECIO	COLOR
Renault Megane	Verde	12.000€	Seat Ibiza	20.000€	Blanco
Ford Focus	Azul	15.000€	Renault Megane	12.000€	Verde
Seat Ibiza	Blanco	20.000€	Ford Focus	15.000€	Azul

Sí es la misma relación ya que el orden de los atributos no es significativo.

Ejemplo3: RELACIÓN COCHES. ¿Está bien formada dicha relación?

<u>MODELO</u>	PRECIO	COLOR
Seat Ibiza	20.000€	Blanco
Renault Megane	12.000€	Verde
Seat Ibiza	20.000€	Blanco
Ford Focus	15.000€	Azul, Blanco

No está bien formada por dos motivos:

- Las tuplas 1 y 3 son iguales
- En la tupla 4, el atributo color no está tomando un único valor, sino dos valores del dominio

Ejemplo4:

RELACIÓN COCHES			RELACIÓN CLIENTES			
<u>MODELO</u>	COLOR	PRECIO	<u>DNI</u>	NOMBRE	APELLIDOS	COCHE
Renault Megane	Verde	12.000€	28954321	Antonio	García	Renault Megane
-----	Azul	15.000€	23542111	Carlos	López	-----
Seat Ibiza	Blanco	20.000€	24232989	Susana	Garrido	Ford Focus



CLAVE AJENA

- La relación Coches no cumple la regla de integridad de entidad ya que la clave primaria está tomando un valor nulo.

- La relación Clientes no cumple la regla de integridad referencial ya que la clave ajena Coche está tomando un valor (Ford Focus) que no concuerda con ningún valor de la clave primaria Modelo.

¿Estaría bien que la clave ajena Coche tome un valor nulo como lo hace en la segunda tupla? Sí porque una clave ajena sí puede tomar un valor nulo.

2.4. RESTRICCIONES SEMÁNTICAS O DE USUARIO

Representan la semántica del mundo real. Éstas hacen que las ocurrencias de los esquemas de la base de datos sean válidos. Los mecanismos que proporciona el modelo para este tipo de restricciones son las siguientes:

- **UNIQUE:**
 - Permite definir claves alternativas o candidatas. Los valores de estos atributos por definición no pueden repetirse.
 - Permite declarar como únicos alguno/s de los atributos que forman una clave primaria para que él/ellos no puedan repetirse en las distintas filas.
 - Cualquier otro atributo que no sea clave candidata puede definirse también como UNIQUE aunque pueda tomar valores nulos ya que los Nulos no son considerados como valores repetidos en MySQL.
- **NOT NULL:** permite declarar si uno o varios atributos no pueden tomar valores nulos.
Ej: PERSONA(Cod_persona, Nombre NOT NULL)
- **CHECK:** restricción de verificación. Permite especificar condiciones que deban cumplir los valores de los atributos. Cada vez que se realice una inserción o una actualización de datos se comprueba si los valores cumplen la condición.
Ej: CATEGORIA(Cod_categ, nombre)
nombre CHECK ("Profesor", "Tutor", "Director")