

GESTIÓN DE BASES DE DATOS

Bases de Datos Relacionales

1. El modelo Relacional.	2
2.1.- Estructura del modelo relacional.	2
1.1.- Dominios y Atributos.	3
1.2.- Relaciones.	3
1.3.- Claves.	4
2.- Restricciones del modelo relacional.	5
2.1.- Inherentes al modelo.	6
2.2.- Semánticas o de usuario.	6
3.- Transformación de un esquema E-R a un esquema relacional.	7
3.1.- Transformación de relaciones N-M.	8
3.2.- Transformación de relaciones 1-N.	9
3.3.- Transformación de relaciones 1:1.	11
3.4.- Transformación de relaciones reflexivas o recursivas.	13
3.5.- Transformación de jerarquías al modelo relacional.	14

1. El modelo Relacional.

El modelo de datos relacional fue desarrollado por Codd⁵. El modelo de Codd persigue al igual que la mayoría de los modelos de datos los siguientes objetivos:

1. **Independencia física de los datos**, esto es, el modo de almacenamiento de los datos no debe influir en su manipulación lógica.
2. **Independencia lógica de los datos**, es decir, los cambios que se realicen en los objetos de la base de datos no deben repercutir en los programas y usuarios que acceden a ella.
3. **Flexibilidad**, para presentar a los usuarios los datos de la forma más adecuada.
4. **Uniformidad**, en la presentación de la lógica de los datos, que son tablas, lo que facilita la manipulación de la base de datos por parte de los usuarios.
5. **Sencillez**, este modelo es fácil de comprender y utilizar por el usuario.

Para conseguir estos objetivos Codd introduce el concepto de **relación (tablas)** como estructura básica del modelo, todos los datos de una base de datos se representa en forma de relaciones cuyo contenido varía en el tiempo.

2.1.- Estructura del modelo relacional.

La **relación** es el elemento básico del modelo relacional y se representa como una tabla, en la que se puede distinguir:

- el nombre de la tabla.
- el conjunto de **columnas** que representan las propiedades de la tabla y que se denominan **atributos**,
- y el conjunto de **filas**, llamadas **tuplas** que contienen los **valores** que toman cada uno de los atributos para cada elemento de la relación.

Una relación tiene una serie de elementos característicos que la distinguen de una tabla:

- ☞ No admiten filas duplicadas.
- ☞ Las filas y las columnas no están ordenadas.
- ☞ La tabla es plana. En el cruce de una fila y una columna solo puede haber un valor.

RELACIÓN ALUMNOS				ATRIBUTOS
NUM_MAT	NOMBRE	APELLIDOS	CURSO	
5496	JUAN	CABELLO	1º BACH-A	TUPLAS
3421	DOLORES	GARCÍA	1º BACH-B	
7622	JESUS	SÁNCHEZ	1º BACH-C	

Los elementos que constituye el modelo relacional son:

1.1.- Dominios y Atributos.

DOMINIO

Se define DOMINIO como el conjunto finito de valores homogéneos (todos del mismo tipo) y atómicos (son indivisibles) que puede tomar cada atributo. Los valores contenidos en una columna pertenecen a un dominio que previamente se ha definido. Todos los dominios tienen un nombre y un tipo de datos asociados. Existen dos tipos de dominios:

- a. **Dominios generales:** son aquellos cuyos valores están comprendido entre un máximo y un mínimo. Por ejemplo: código postal, formado por todos los números enteros positivos de cinco cifras.
- b. **Dominios restringidos:** Son los que pertenecen a un conjunto de valores específicos. *Por ejemplo: sexo. Solamente puede tomar los valores H y M.*

ATRIBUTO

Se define ATRIBUTO como el papel o rol que desempeña un dominio en una relación. Representa el uso de un dominio para una determinada relación. El atributo aporta un significado semántico a un dominio. Por ejemplo, en la relación ALUMNOS podemos considerar los siguientes atributos y dominios:

- Atributo NUM_MAT. Dominio: conjunto de enteros formados por 4 dígitos.
- Atributo NOMBRE. Dominio: conjunto de 15 caracteres.
- Atributo APELLIDOS. Dominio: conjunto de 20 caracteres.
- Atributo CURSO. Dominio: conjunto de 7 caracteres.

1.2.- Relaciones.

Las relaciones se representan mediante una tabla con filas y columnas. Un SGBD solo necesita que el usuario pueda percibir la BD como un conjunto de tablas.

En el modelo relacional las relaciones se utilizan para almacenar información sobre los objetos

que se representan en la BD. Se representa como una tabla bidimensional en la que las filas corresponden a registros individuales y las columnas a los campos o atributos de esos registros. La relación está formada por:

- **Atributos (columnas).** Se trata de cada una de las columnas de la tabla. Las columnas tienen un nombre y pueden guardar un conjunto de valores. Una columna se identifica siempre por su nombre, nunca por su posición. El orden de las columnas en una tabla es irrelevante.
- **Tuplas (filas).** Cada tupla representa una fila de la tabla. En la siguiente tabla vemos que aparecen tres tuplas o filas, y cuatro atributos (num_mat, nombre, apellidos, curso).

RELACIÓN ALUMNOS				ATRIBUTOS
NUM_MAT	NOMBRE	APELLIDOS	CURSO	
5496	JUAN	CABELLO	1º BACH-A	TUPLAS
3421	DOLORES	GARCIA	1º BACH-B	
7622	JESUS	SÁNCHEZ	1º BACH-C	

De las tablas se derivan los siguientes conceptos:

- **Cardinalidad.** Es el número de filas de la tabla. En el ejemplo anterior, es TRES.
- **Grado.** Es el número de columnas de la tabla. En el ejemplo anterior, es CUATRO.
- **Valor.** Está representado por la intersección entre una fila y una columna. Por ejemplo, en la tabla anterior, son valores “CABELLO” “JUAN”, 7622...
- **Valor null.** Representa la ausencia de información.

Las relaciones tienen las siguientes características:

- Cada relación tiene un nombre y este es distinto de los demás.
- Los valores de los atributos son atómicos: en cada tupla, cada atributo toma un solo valor.
- No hay dos atributos que se llamen igual.
 - El orden de los atributos es irrelevante; no están ordenados.
 - Cada tupla es distinta de las demás; no hay tuplas duplicadas.
- Al igual que los atributos, el orden de las tuplas es irrelevante; las tuplas no están ordenadas.

1.3.- Claves.

En una relación no hay tuplas repetidas; se identifica de un modo único mediante los valores de sus atributos. Toda fila debe estar asociada con una clave que permite identificarla. A veces las filas se pueden identificar con un mismo atributo, pero otras veces es necesario recurrir a más de un atributo.

La **clave candidata** de una relación es el conjunto de atributos que **identifica de forma única y mínima** cada tupla de la relación. **Siempre hay una clave candidata.**

Una relación puede tener más de una clave candidata entre las cuales se distinguen:

☛ **Clave primaria o principal:** aquella clave candidata que el usuario escoge para identificar las tuplas de la relación. No puede tener valores nulos.

☛ **Clave alternativa:** aquellas claves candidatas que no han sido escogidas como clave primaria.

- ☐ La **clave ajena** de una relación R1 es el conjunto de atributos cuyos valores han de coincidir con los valores de la clave primaria de otra relación R2.

Ejemplo:

TABLA DEPARTAMENTOS

CLAVE PRIMARIA

CLAVE CANDIDATA

Nº DEPARTAMENTO	NOMBRE DEPARTAMENTO	PRESUPUESTO
D1	MARQUETING	1000
D2	DESARROLLO	1200
D3	INVESTIGACIÓN	5000

TABLA EMPLEADOS

CLAVE PRIMARIA

CLAVE AJENA

Nº EMPLEADO	APELLIDOS	Nº DEPARTAMENTO	SALARIO
E1	LOPEZ	D1	500
E2	FERNANDEZ	D3	1000
E3	GARCÍA	D2	1200

2.- Restricciones del modelo relacional.

En todos los modelos de datos existen restricciones que a la hora de diseñar una base de datos se han de tener en cuenta. Los datos almacenados en la base de datos han de adaptarse a las estructuras

impuestas por el modelo y deben cumplir una serie de reglas para garantizar que son correctas.

Los tipos de restricciones son:

2.1.- Inherentes al modelo.

Indican las características propias de una relación que ha de cumplirse obligatoriamente, y que diferencian una relación de una tabla:

- No hay dos tuplas iguales,
- El orden de la tuplas y los atributos no es relevante,
- Cada atributo sólo puede tomar un único valor del dominio al que pertenece y,
- Ningún atributo que forme parte de la clave primaria de una relación puede tomar un valor nulo.

2.2.- 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 proporcionan el modelo para este tipo de restricciones son los siguientes:

- **Restricción de clave primaria (PRIMARY KEY)**, permite declarar uno o varios atributos como clave primaria de una relación.
- **Restricción de unicidad (UNIQUE)**, permite definir claves alternativas. Los valores de los atributos no pueden repetirse.
- **Restricción de obligatoriedad (NOT NULL)**, permite declarar si uno o varios atributos no pueden tomar valores nulos. (Por ejemplo: podríamos declarar que el atributo “Apellidos” de la tabla anterior “Empleados”, no pueda dejarse en blanco).
- **Integridad referencial o restricción de clave ajena**, la integridad referencial indica que los valores de clave ajena en la relación hijo se corresponden con los de claves primarias en la relación padre. (Por ejemplo: que no pueda aparecer en la tabla “Ventas” el código de un producto si dicho producto no se encuentra en la tabla “Productos”).

Además de definir las claves ajenas hay que tener en cuenta las operaciones de borrado y actualización que se realizan sobre las tuplas de la relación referencial. Las **posibilidades** son las siguientes:

Borrado y/o modificación en cascada. (Cascade). El borrado o modificación de una tupla en la relación padre (relación con la clave primaria) ocasiona un borrado o modificación de las tuplas relacionadas con la relación hija (relación que contiene la clave ajena).

Borrado y/o modificación restringido. (Restrict). En este caso no es posible realizar el

borrado o la modificación de las tuplas de la relación padre si existen tuplas relacionadas en la relación hija. Es decir, no podría borrar un departamento que tiene empleados.

Borrado y/o modificación con puesta o nulos (Set null). Esta

La restricción permite poner la clave ajena en la tabla referenciada a NULL si se produce el borrado o modificación en la tabla primaria o padre. Así pues, si se borra un departamento, a los empleados de ese departamento se asignará NULL en el atributo NUMDEPT.

Borrado y/o modificación con puesta o valor por defecto (Set default). En este caso, el valor que se pone en la claves ajenas de la tabla referenciada es un valor por defecto que se habrá especificado en la creación de la tabla.

- **Restricción de verificación (check)** Esta restricción permite especificar condiciones que deben 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. Rechaza la operación si no se cumple.

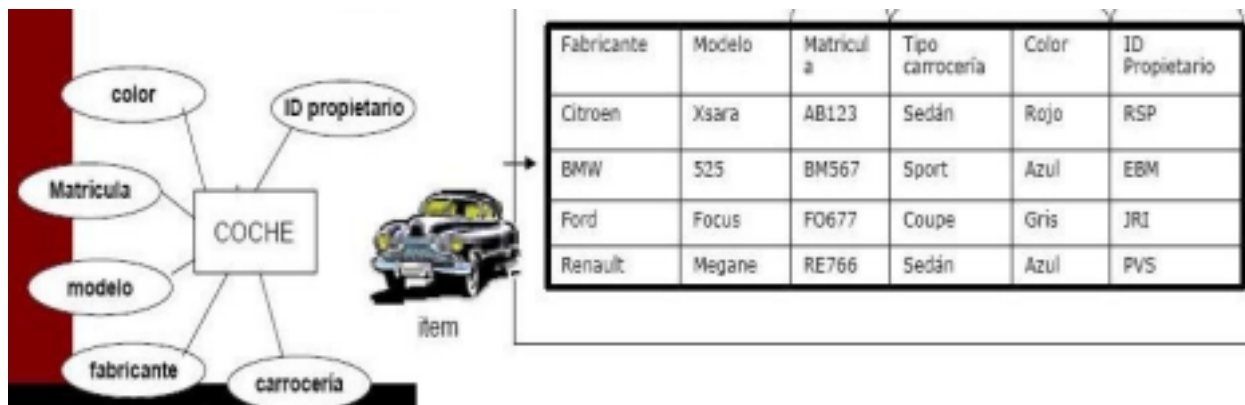
→ Ejemplo: que el campo PRECIO_VENTA sea mayor que 0.

→ Ejemplo: Que el campo EDAD sea mayor que 0, y (por ejemplo) menor que 120.

3.- Transformación de un esquema E-R a un esquema relacional.

Una vez obtenido el esquema conceptual mediante el modelo E-R hay que definir el modelo lógico de datos. Las reglas básicas para transformar un esquema conceptual E-R a un esquema relacional son las siguientes:

- ☛ Toda entidad se transforma en una tabla.
- ☛ Todo atributo se transforma en columnas dentro de una tabla.
- ☛ El identificador único de la entidad se convierte en clave primaria.

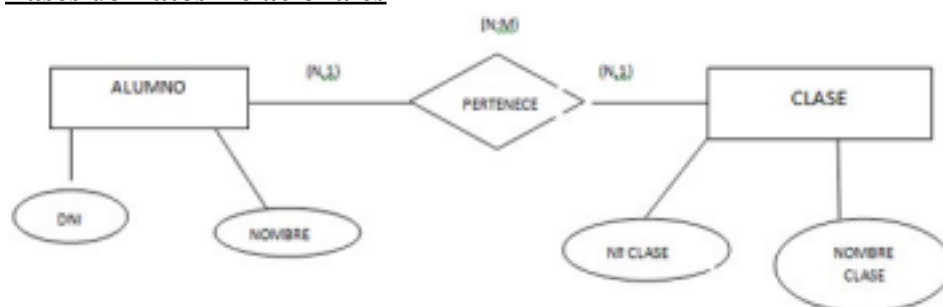


¿Cómo hacemos esto?. Dependiendo de las relaciones que nos encontremos:

3.1.- Transformación de relaciones N-M.

Toda relación N:M se transforma en una tabla que tendrá como clave primaria la concatenación de los atributos que asocia .

Bases de Datos Relacionales



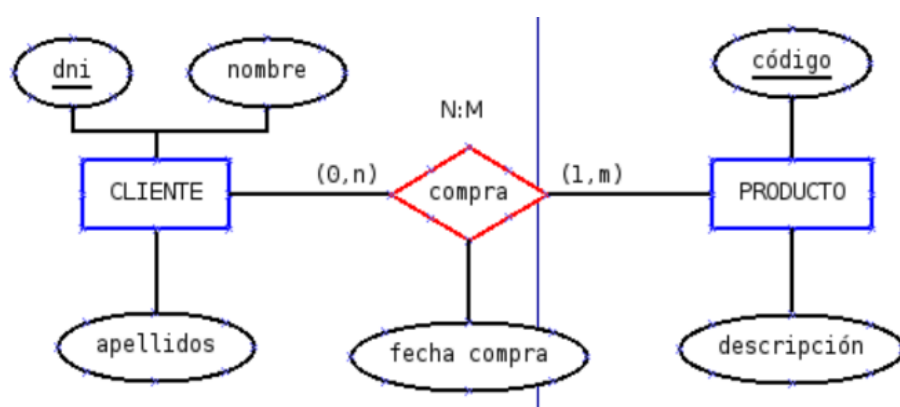
Para este modelo de entidad-relación el paso a tablas quedaría de la siguiente forma:

Tabla alumno	DNI(clave primaria)	nombre
Tabla clase	Nº clase (clave primaria)	Nombre clase
Tabla pertenece	DNI (clave foránea)	Nº clase (clave foránea)

}
Clave primaria

Otro ejemplo: Supongamos el siguiente **modelo entidad-relación**.

En este caso la relación “**compra**” se transforma en una nueva tabla cuya clave primaria estará



formada por los atributos dni, que es la clave primaria de cliente, y código, que es la clave primaria de producto. Además tendrá como campo fecha compra, ya que este atributo forma parte de la

relación.

El **modelo relacional** quedaría de la siguiente forma (en negrita las claves primarias):

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

3.2.- Transformación de relaciones 1-N.

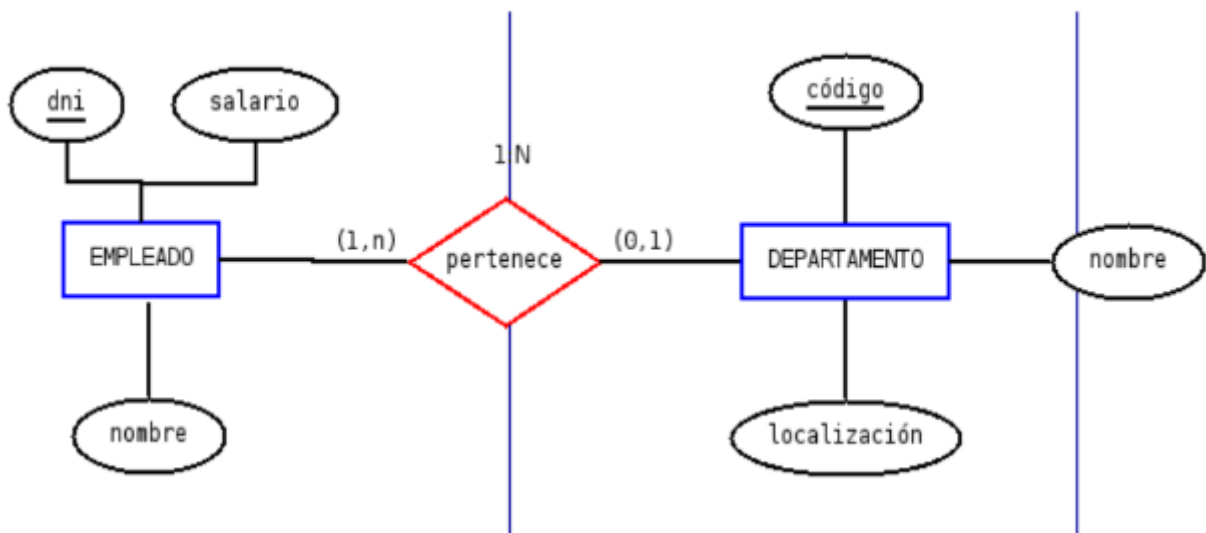
Existen dos soluciones:

- ☛ **Transformarla en una tabla.** Se hace como si se tratara de una relación N:M. Esta solución se realiza cuando se prevé que en un futuro la relación se convertirá en N:M y cuando la relación tiene atributos propios. La clave de esta tabla es la de la identidad del lado muchos. También se aplica cuando tenemos cardinalidad (0,1) y (1,N).

Ejemplo:

En el siguiente **modelo entidad-relación** un empleado pertenece a un único departamento, y un departamento tiene 1 o más empleados.

Imaginemos ahora que pudiera darse el caso de que hubiera empleados que no pertenecieran a ningún departamento.



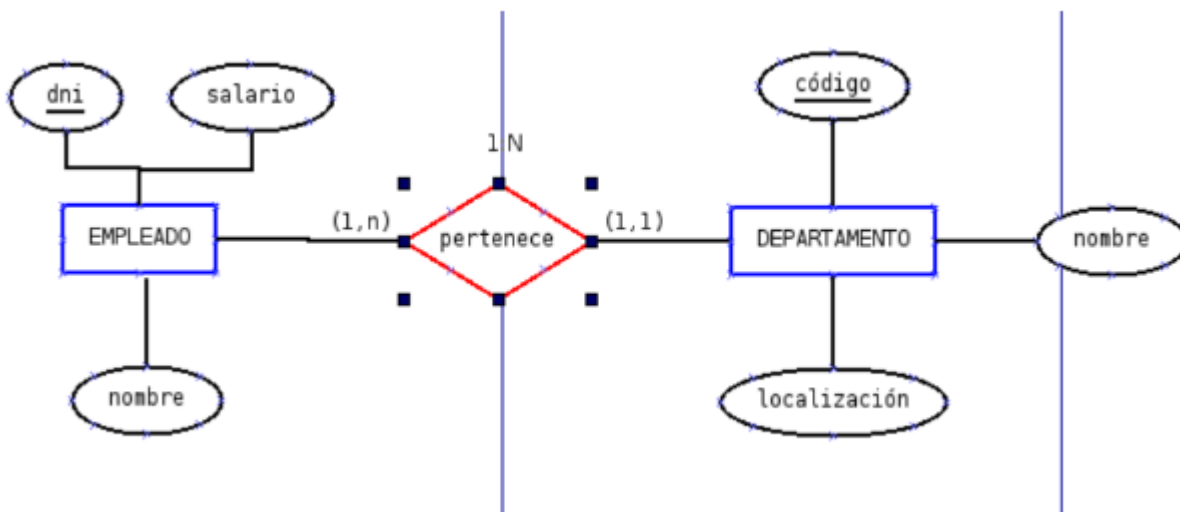
En este caso la entidad que participa con cardinalidad máxima 1, DEPARTAMENTO, también lo hace con cardinalidad mínima 0, ya que puede haber empleados que no pertenezcan a ningún departamento.

Así pues, se crea una nueva tabla formada por dni de EMPLEADO y código de DEPARTAMENTO. En esta nueva tabla dni de EMPLEADO será la clave primaria. El **modelo relacional** quedaría de la siguiente forma:

- EMPLEADO(**dni**, nombre, salario)
- DEPARTAMENTO(**código**, nombre, localización)
- PERTENECE(**dni_empleado**, código_departamento)

☛ **Propagar la clave.** Este caso se aplica cuando la cardinalidad es obligatoria, es decir, cuando tenemos cardinalidad (1,1) y (0,n). Se propaga el atributo principal de la entidad que tiene de cardinalidad máxima 1 a la que tiene de la cardinalidad máxima N, desapareciendo el nombre de la relación.

Ejemplo 1: Veamos ahora el caso de una relación 1:N. En el siguiente **modelo entidad-relación** un empleado pertenece a un único departamento (**debe pertenecer a uno obligatoriamente**), y un departamento tiene 1 o más empleados.



En este caso se propaga el atributo código de departamento a la tabla EMPLEADO. El **modelo relacional** quedaría de la siguiente manera:

EMPLEADO(**dni**, nombre, salario, código_departamento)

DEPARTAMENTO(**código**, nombre, localización)

3.3.- Transformación de relaciones 1:1.

Se tienen en cuenta las cardinalidades de las entidades que participan. Existen dos soluciones:

1.- Transformarla en una tabla. Si las entidades poseen cardinalidades (0,1), la relación se convierte en una tabla.

Ejemplo1:

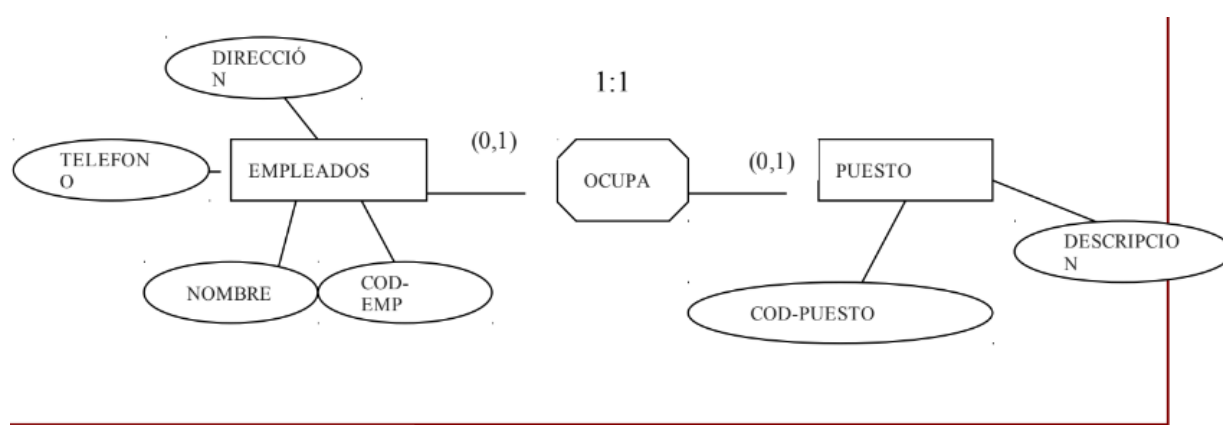
Supongamos que disponemos de una entidad EMPLEADOS, que contiene atributos relativos a los empleados de una empresa.

Supongamos también que, disponemos de una entidad PUESTO, que contiene atributos relativos a un determinado puesto de trabajo.

Supongamos también que, un puesto es único. Es decir, un determinado puesto solamente puede estar ocupado por una única persona. Además, este puesto puede estar o no ocupado.

Algunos empleados ocuparán puestos, pero otros son simples empleados sin puesto asignado.

Así tenemos:

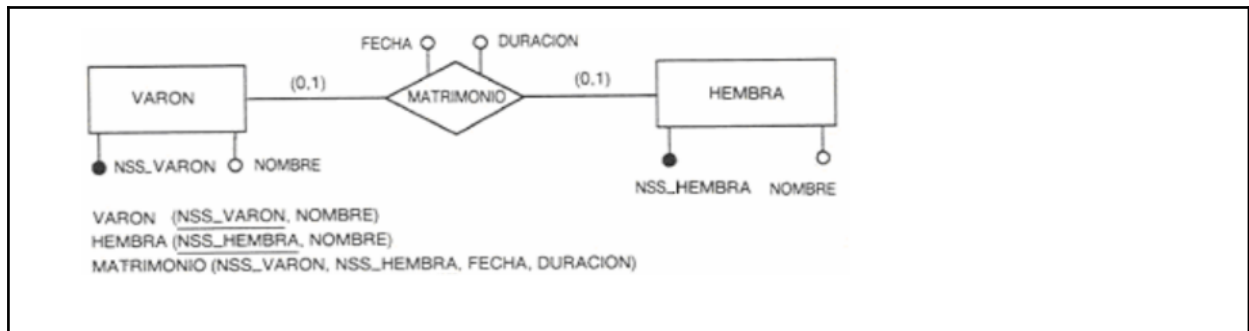


EMPLEADOS: (cod_empleado, nombre, dirección, teléfono).

PUESTO: (cod_puesto, descripción).

OCUPA: (cod_empleado, cod_puesto).

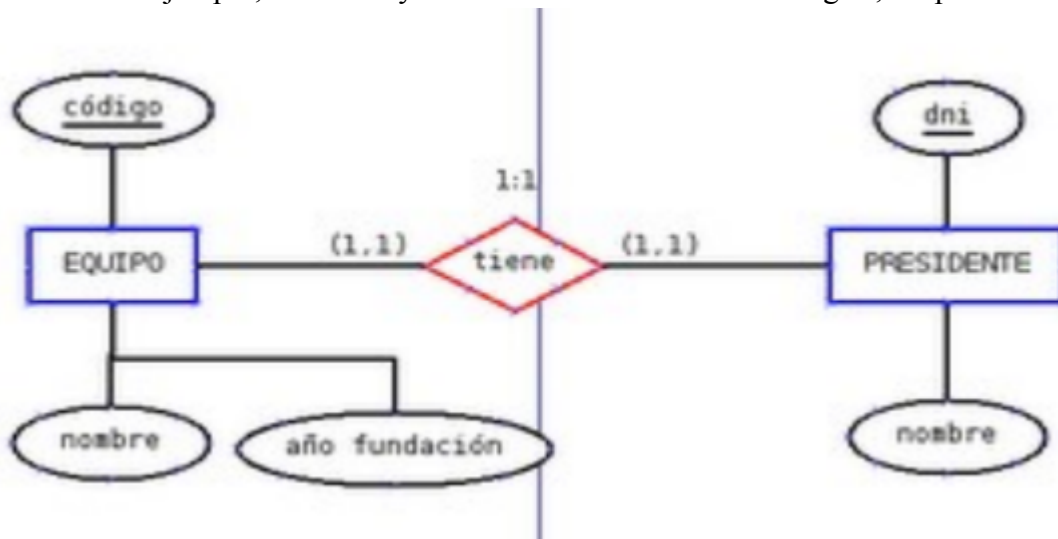
Ejemplo2:



2.- Propagar la clave. Si una de las entidades posee cardinalidad (0,1) y la otra (1,1) conviene propagar la clave de la entidad con cardinalidad (1,1) a la tabla resultante de la entidad de cardinalidad (0,1). Si ambas entidades poseen cardinalidades (1,1) se puede propagar la clave de cualquiera de ella a la tabla resultante de la otra. En este caso también se puede añadir los atributos de una entidad a otra, de donde resulta una única tabla con todos los atributos de las entidades y de la relación si los hubiera, eligiendo como clave primaria una de las dos.

Ejemplo: Veamos ahora el caso de una relación 1:1 a través del siguiente ejemplo. En el siguiente **modelo entidad-relación** un equipo de fútbol tiene a un único presidente y un presidente preside a un único club de fútbol.

En este ejemplo, tal y como dicen las reglas, podemos propagar



la clave de cualquier tabla a la tabla resultante de la otra. Es decir, tenemos dos opciones, o mover la clave de PRESIDENTE a EQUIPO o mover la clave de EQUIPO a PRESIDENTE. El **modelo relacional** podría quedar de cualquiera de las dos formas siguientes:

- EQUIPO(**código**, nombre, año_fundación)
 - PRESIDENTE(**dni**, nombre, código_equipo)
-
- EQUIPO(**código**, nombre, año_fundación, dni_presidente)
 - PRESIDENTE(**dni**, nombre)

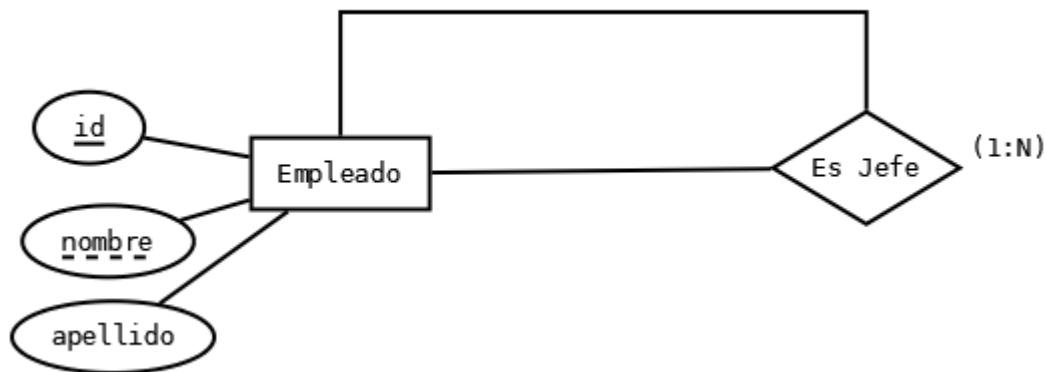
3.4.- Transformación de relaciones reflexivas o recursivas.

En el caso de que la relación sea:

- ☛ **Relación 1:1.** La clave de la entidad se repite, con lo que la tabla resultante tendrá dos veces ese atributo, una como clave primaria y otra como clave ajena de la misma.

Ejemplo: imaginemos una tabla de PERSONAS. Tenemos una relación “es cónyuge”. Es una relación 1:1. Cada fila contendrá (por ejemplo) el DNI y todos los datos de una persona, siendo el DNI la llave primaria, y además, la llave primaria se repetirá. Es decir, otro campo DNI_CONYUGE que será el DNI del cónyuge.

- ☛ **Relación 1:N.** Tenemos dos casos:
Aquel en el que la entidad muchos es obligatorio, se procede como en el caso (1,1).



Ejemplo: “JEFE” 1 a N. - La relación ‘Jefe’, es una relación 1:N pero no es obligatoria. Por tanto, la Tabla EMPLEADOS queda con un atributo más, que es el número de empleado del jefe del empleado. Le ponemos como nombre JEFE.
EMPLEADO (ID, nombre, Apellido, Jefe)

- Si no es obligatoria se crea una nueva tabla cuya clave será la de la entidad y además se propaga la clave a la nueva tabla como clave ajena.

☛ **Relación N:M.** Se trata igual que las relaciones binarias. Es decir, se crea una tabla nueva.

3.5.- Transformación de jerarquías al modelo relacional.

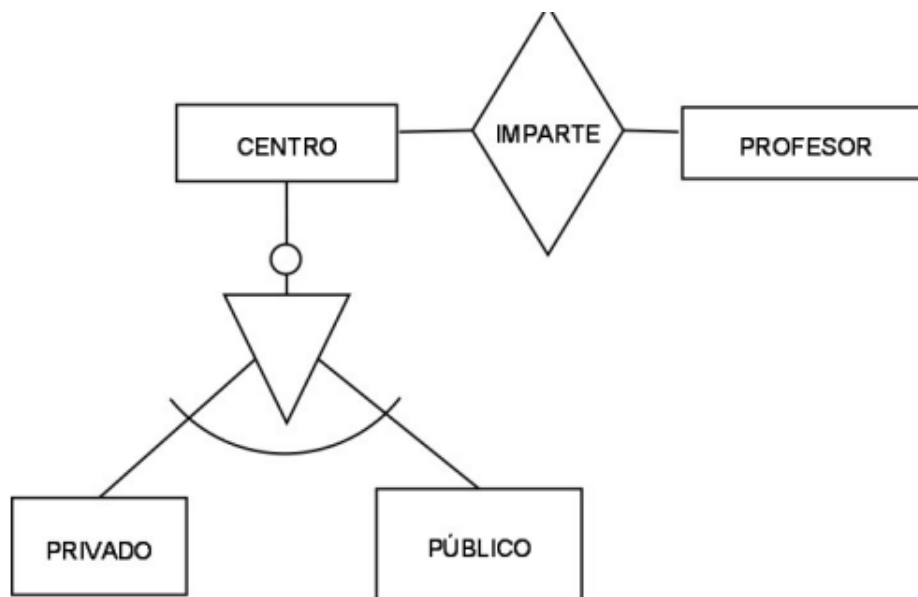
El modelo relacional no dispone de mecanismo para la representación de las relaciones jerárquicas; así pues, estas relaciones se tienen que eliminar.

Para pasar estas relaciones al modelo relacional se pueden aplicar distintas reglas. Nosotros veremos la siguiente:

- ☛ **Integrar todas las entidades en una única eliminando a los subtipos.** Esta nueva entidad contendrá todos los atributos del supertipo, todos los de los subtipos, y los atributos discriminativos para distinguir a que subentidad pertenece cada atributo. Esta regla puede aplicarse a cualquier tipo de jerarquía.
- ☛ La gran ventaja es la simplicidad, pues todo se reduce a una entidad.
- ☛ El gran inconveniente es que se generan demasiados valores nulos en los atributos opcionales propios de cada entidad.

Ejemplo: Consideramos los profesores que imparten clase en dos tipos de centro educativos: públicos y privados.

- Un profesor puede impartir clase en varios centros, sean públicos o privados.
- Consideramos la asignatura como atributo de la relación entre profesores y centro profesores –imparten - clase
- Los centros educativos sólo pueden ser de estos dos tipos y un centro público no puede ser privado.
- Los atributos específicos para los públicos son: el presupuesto y los servicios, y para los privados, la organización y la cuota.
- De los profesores se conoce su código, su nombre, su dirección, teléfono y localidad. Y de los centros: el código, el nombre, la dirección y el nº de alumno.



Profesores: (Código-profe, nombre, dirección, teléfono, localidad)

Imparte: (Código-profe, código-centro, asignatura)

Centros: (código-centro, nombre, dirección, número alumno, organización, cuota, presupuesto, servicio)

Este modelo E-R no está completo. Faltan los atributos, la cardinalidad, etc. Complétalo tú como actividad.