

GESTIÓN DE BASES DE DATOS

Bases de Datos Relacionales

IES GONZALO NAZARENO

UNIDAD DIDACTICA 2: BASES DE DATOS RELACIONALES

Índice de contenido

1. El modelo Entidad-Relación (ER).....	3
1.1.- Elementos del modelo E-R.....	3
1.1.1.- Entidades.....	3
1.1.1.1.- Tipos de Entidades.....	3
1.1.1.2.- Componentes de las entidades.....	4
1.1.2.- Relaciones.....	5
1.1.2.1.- Grado de las relaciones.....	5
1.1.2.2.- Cardinalidad de las relaciones.....	7
1.2.- Generalización y jerarquías de generalización.....	8
1.2.1.- Parcial/Total y Exclusiva/Solapada.....	8
2. El modelo Relacional.....	9
2.1.- Estructura del modelo relacional.....	10
2.1.1.- Dominios y Atributos.....	10
2.1.2.- Relaciones.....	11
2.1.3.- Claves.....	12
2.2.- Restricciones del modelo relacional.....	13
2.2.1.- Inherentes al modelo.....	13
2.2.2.- Semánticas o de usuario.....	13
2.3.- Transformación de un esquema E-R a un esquema relacional.....	16
2.3.1.- Transformación de relaciones N-M.....	16
2.3.2.- Transformación de relaciones 1-N.....	17
2.3.3.- Transformación de relaciones 1-1.....	20
2.3.4.- Transformación de relaciones reflexivas o recursivas.....	22
2.3.5.- Transformación de jerarquías al modelo relacional.....	24

UD02: Bases de Datos Relacionales.

1. El modelo Entidad-Relación (ER).

El modelo entidad-relación (E-R) fue propuesto por Peter Chen, en 1976 para la representación conceptual de los problemas del mundo real. Es un modelo muy extendido y potente para la representación de los datos; es la referencia que se utiliza para el diseño de modelos relacionales. Se simboliza haciendo uso de gráficos en lo que se representa las entidades y las relaciones entre ellas.

1.1.- Elementos del modelo E-R.

1.1.1.- Entidades.

Son objetos del mundo real que tienen interés para la empresa. Cada entidad se identifica por su nombre. Se representa utilizando rectángulos.

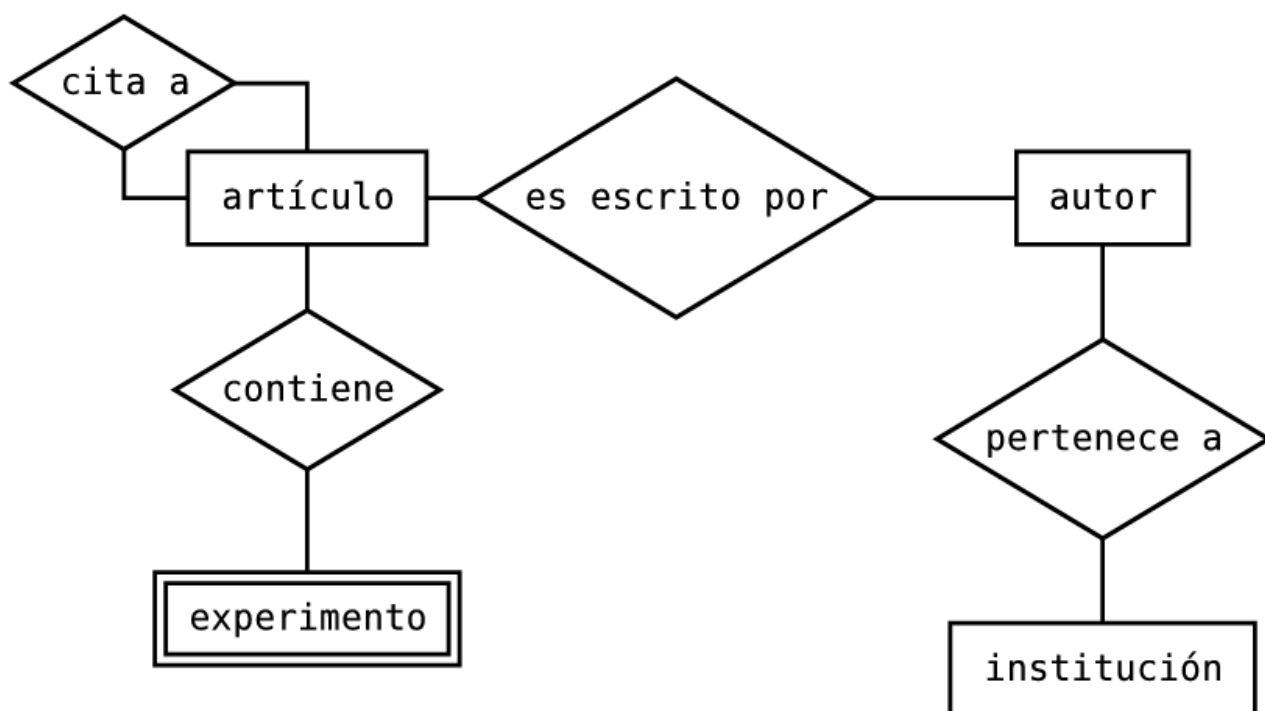
Un ejemplo sería la entidad banco, donde se recogerían los datos relativos a ese banco, como puede ser el nombre, el número de sucursal, la dirección, etc.

1.1.1.1.- Tipos de Entidades.

Hay dos tipos de entidades:

- **Entidad fuerte:** Es aquella que no depende de otra para su existencia.
Ejemplo: la entidad **ALUMNOS** perteneciente a un centro escolar, o la entidad **CLIENTES** de un banco.
- **Entidad débil:** Es aquella que necesita de otra entidad para existir. Se representa mediante un rectángulo doble.
Ejemplo: la entidad **NOTAS** necesita a la entidad **ALUMNOS**, porque sin alumnos no hay notas.

Ejemplo: en el siguiente gráfico podemos ver varias entidades fuertes, representadas por rectángulos (artículo, autor, institución). También podemos ver una entidad débil, representada por un rectángulo doble (experimento).

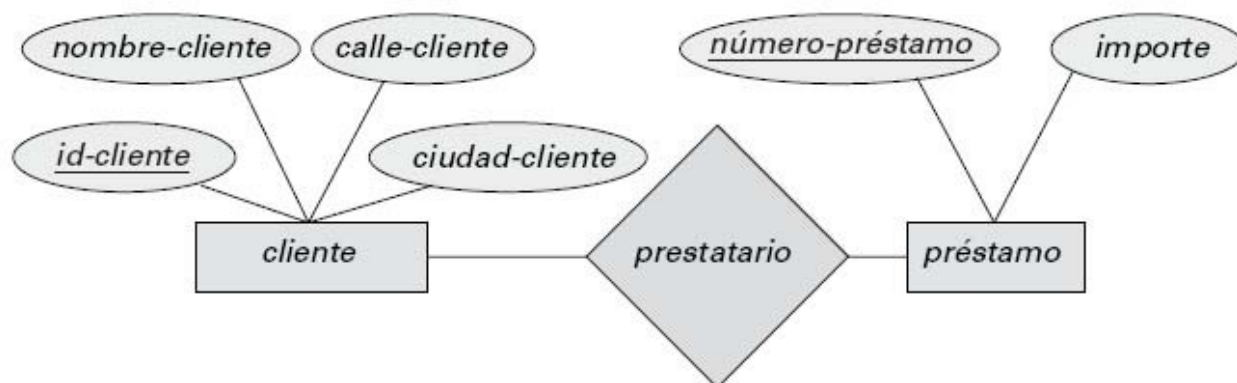


1.1.1.2.- Componentes de las entidades.

1. **Atributos:** Son las unidades de información que describe propiedades de entidades. El conjunto de valores permitidos para cada atributo se llama dominio. Los atributos se representan mediante una elipse con el nombre dentro.

Por ejemplo: en la siguiente imagen, la entidad CLIENTE tiene los siguientes atributos: id-cliente, nombre-cliente, calle-cliente, ciudad-cliente. La entidad PRÉSTAMO tiene los siguientes atributos: número-préstamo, importe.

Ejemplo: el dominio del campo calle-cliente será el formado por todas las cadenas de longitud 20 caracteres.



2. **Clave primaria o principal (Primary key):** Es el conjunto de atributos que identifica de forma única una entidad. No puede contener valores nulos.

La clave primaria ha de ser sencilla de crear, y no ha de variar en el tiempo.

Se representa subrayando el atributo o atributos que forman la clave.

Por ejemplo: en la imagen anterior, la clave primaria de la entidad CLIENTE es id-cliente. La clave primaria de la entidad PRÉSTAMOS es número-préstamo. Tened en cuenta que la clave primaria puede estar formada por más de un atributo.

3. **Clave ajena o foránea (Foreign key):** Es el atributo o conjunto de atributos de una entidad que forman la clave primaria de otra entidad. Las claves ajenas van a representar las relaciones entre tablas.

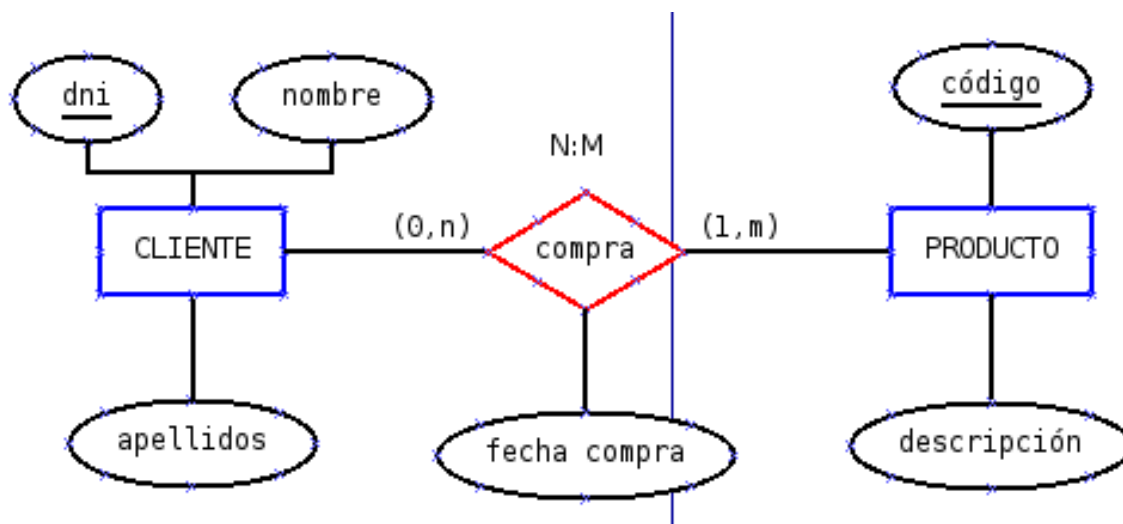
1.1.2.- Relaciones.

Una relación es la asociación entre dos o más entidades. Tienen nombre de verbo y se representa mediante un rombo.

1.1.2.1.- Grado de las relaciones.

Se define el grado de una relación como el número de entidades que participan en la misma. **Las relaciones en las que participan dos entidades se denominan binarias o de grado dos; si participan tres, serán ternarias o de grado tres.** Los conjuntos de relaciones pueden tener cualquier grado. **Lo ideal es tener relaciones binarias.**

La siguiente imagen muestra una relación binaria, de nombre COMPRA (entre las entidades CLIENTE y PRODUCTO):

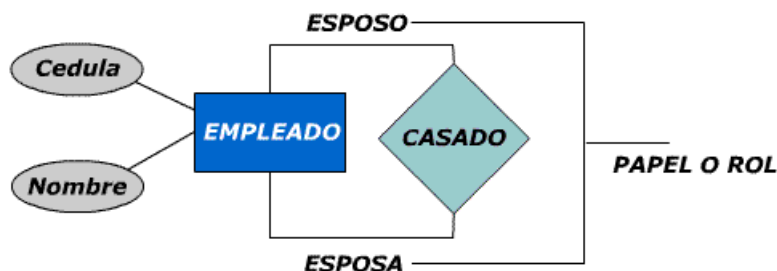


La siguiente imagen muestra una relación ternaria de nombre TRABAJA-EN (entidades EMPLEADO, TRABAJO y SUCURSAL):



Las relaciones en las que solo participan una entidad se llama de grado uno o anillo: una entidad se relaciona consigo misma; se les llama relaciones reflexivas o recursivas.

En la siguiente imagen se muestra una relación unaria, de nombre CASADO (en la que participa una única entidad de nombre EMPLEADO).



1.1.2.2.- Cardinalidad de las relaciones.

Las relaciones, en principio binarias, pueden involucrar a un número distinto de instancias de cada entidad. Así, son posibles tres tipos de cardinalidades:

- Relaciones **de uno a uno**: una instancia de la entidad A se relaciona con una y solamente una de la entidad B¹.
- Relaciones **de uno a muchos**: cada instancia de la entidad A se relaciona con varias instancias de la entidad B^{2 3}.
- Relaciones **de muchos a muchos**: cualquier instancia de la entidad A se relaciona con cualquier instancia de la entidad B⁴.

La cardinalidad de una entidad sirve para conocer su grado de participación en la relación. La representamos entre paréntesis indicando los valores mínimo y máximo. Los valores para la cardinalidad son: (0,1), (1,1), (0, n), (1, n) y (n, m).

Ejemplo: basándonos en el siguiente diagrama E-R.

A un cliente le corresponden 1 o n productos. Es decir, un cliente compra uno o muchos productos (en principio no tiene sentido que un cliente compre 0 productos).

Al mismo tiempo, un producto puede ser comprado por 0 a n clientes. Es decir, un mismo producto (un litro de aceite de oliva virgen extra “La Española” puede ser comprado por ningún

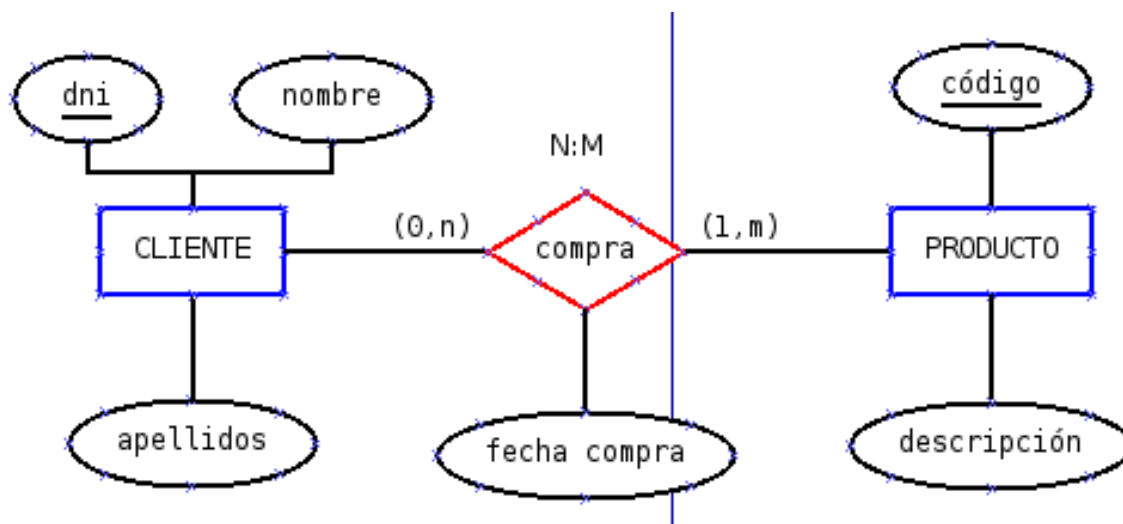
1 Es decir, a cada elemento de la primera entidad le corresponde sólo uno de la segunda entidad, y viceversa. Por ejemplo, un cliente de hotel ocupa una habitación, o un grupo de alumnos pertenece a un aula, y a esa aula sólo asiste ese grupo de alumnos.

2 Es decir, a cada elemento de la primera entidad le corresponde uno o más elementos de la segunda entidad, y a cada elemento de la segunda entidad le corresponde uno solo de la primera entidad. Por ejemplo, un proveedor suministra muchos artículos. Y un artículo es suministrado por solo un proveedor.

3 También se presenta el caso contrario: N:1. Es decir, relaciones de muchos a uno. Es el mismo caso que las de uno a muchos.

4 Es decir, a cada elemento de la primera entidad le corresponde uno o más elementos de la segunda entidad, y a cada elemento de la segunda entidad le corresponden uno o más elementos de la primera entidad. Por ejemplo, un vendedor vende muchos artículos, y un artículo es vendido por muchos vendedores.

cliente o por muchos clientes).

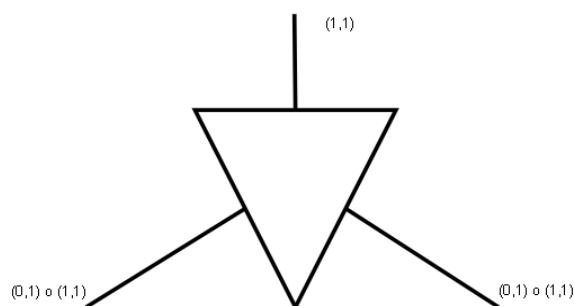


1.2.- Generalización y jerarquías de generalización.

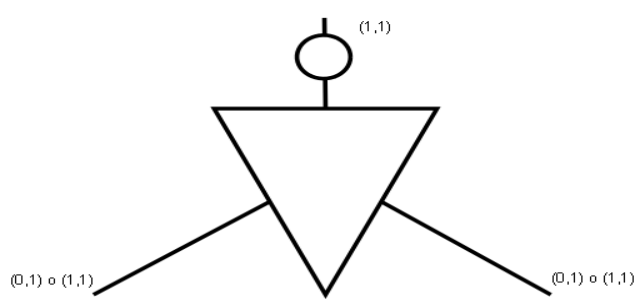
Las generalizaciones proporcionan un mecanismo de abstracción que permite especializar una entidad que se denominará supertipo en subtipos, o lo que es lo mismo, generalizar los subtipos en el supertipo. Una generalización se identifica si encontramos una serie de atributos comunes a un conjunto de entidades, y unos atributos específicos que identificaran unas características. Los atributos comunes describirán el supertipo y los particulares los subtipos. Una de las características más importante de la jerarquía es la herencia, por la que los atributos de un supertipo son heredados por sus subtipos.

1.2.1.- Parcial/Total y Exclusiva/Solapada.

La generalización es total si no hay ocurrencia en el supertipo que no pertenezca a ninguno de los subtipos. En otro caso la generalización sería parcial.

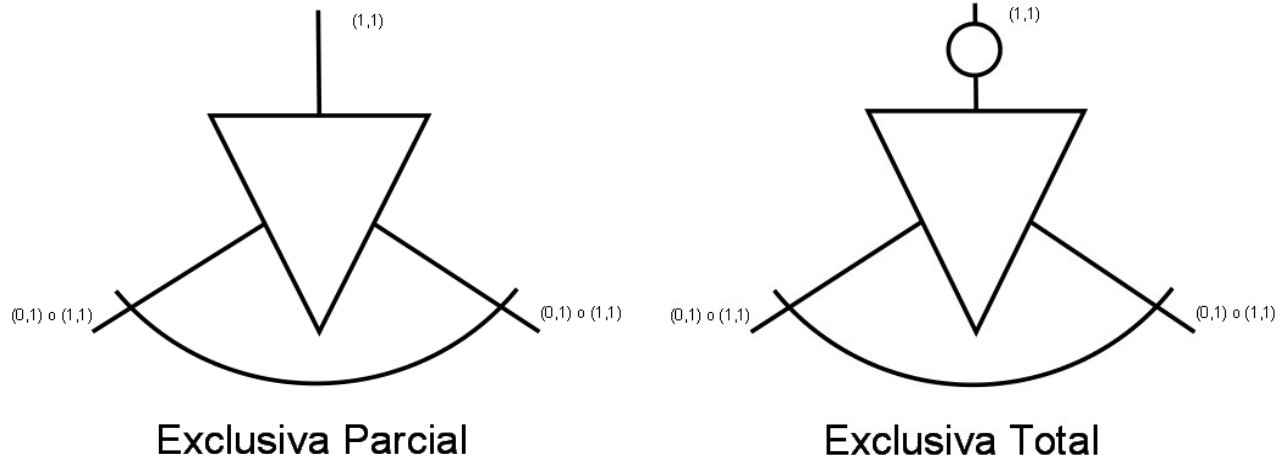


Solapada Parcial

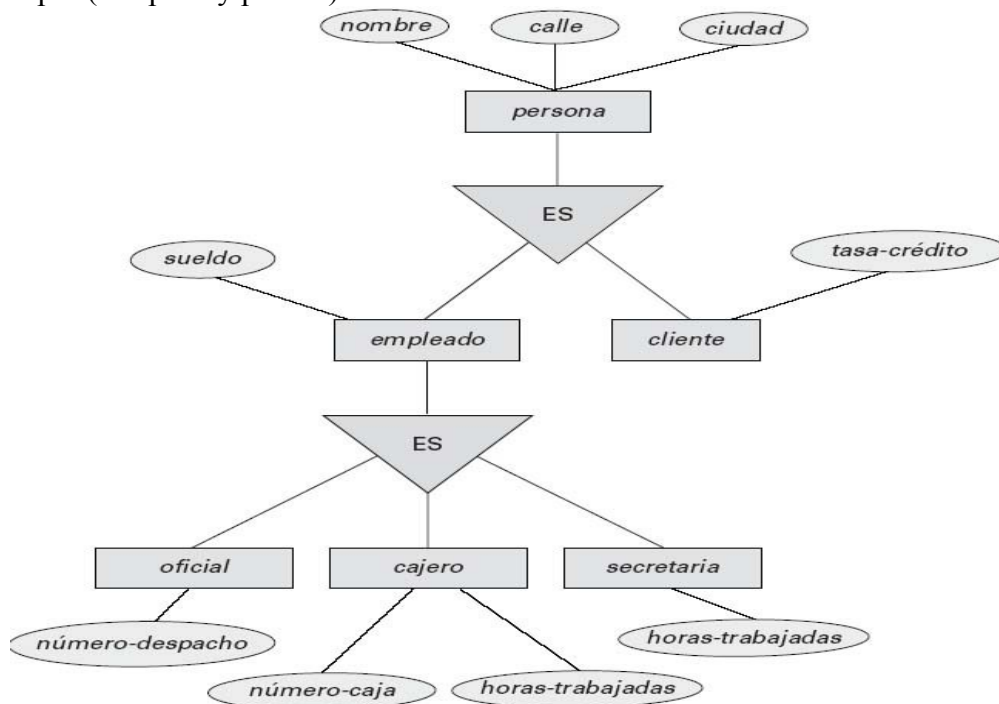


Solapada y Total

La generalización será exclusiva si una ocurrencia no puede aparecer en varios subtipos a la vez.



Ejemplo (solapada y parcial):



2. 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:

⁵Para saber más sobre Edgar Frank Codd: http://es.wikipedia.org/wiki/Edgar_Frank_Codd

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

Los elementos que constituye el modelo relacional son:

2.1.1.- 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 pertenece a un conjunto de valores específico. 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.

2.1.2.- 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	GARCÍA	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.** Esta representado por la intersección entre una fila y 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.

2.1.3.- 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

2.2.- 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:

2.2.1.- 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.

2.2.2.- 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). EJEMPLO: En el caso de empleados y departamentos, si se borra un departamento de la tabla TDEPART se borrarán los empleados que pertenecen a ese departamento. Igualmente ocurrirá si se modifica el NUMDEPT de la tabla TDEPART: esa modificación se arrastra a los empleados que pertenezcan a ese departamento.

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 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.

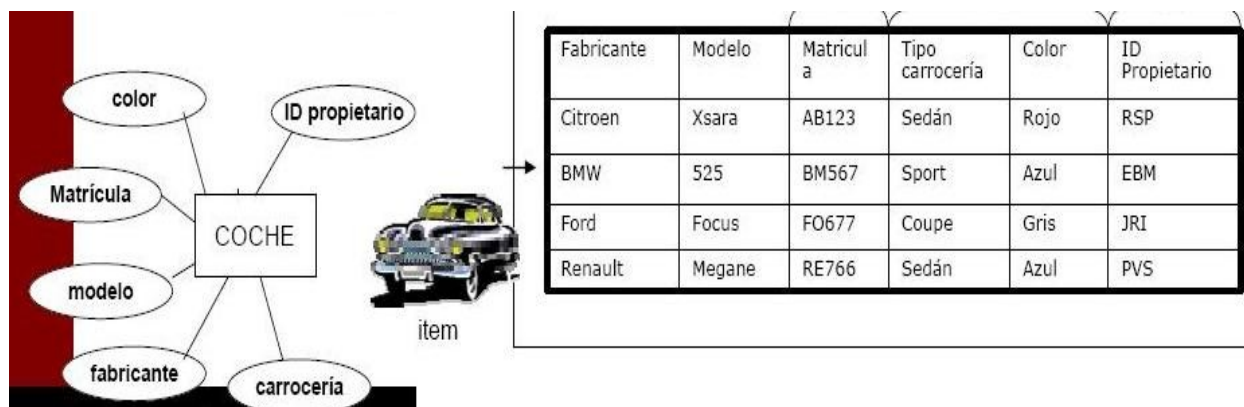
Borrado y/o modificación con puesta o valor por defecto (Set default). En este caso, el valor que se pone en la claves ajenas de la tabla referenciada es un valor por defecto que se habrá especificado en la creación de la tabla.

- **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.

2.3.- 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:

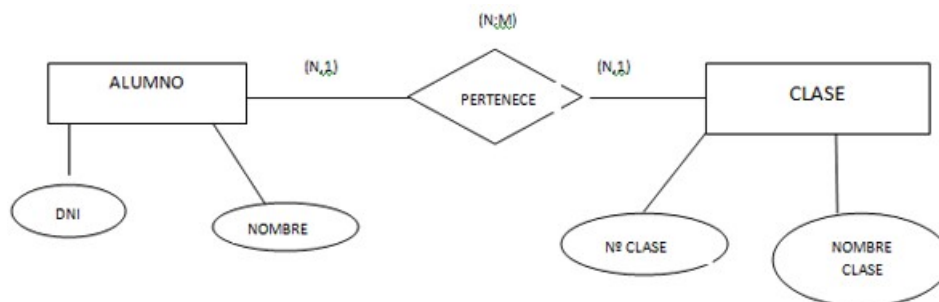
- Toda entidad se transforma en una tabla.
- Todo atributo se transforma en columnas dentro de una tabla.
- El identificador único de la entidad se convierte en clave primaria.



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

2.3.1.- Transformación de relaciones N-M.

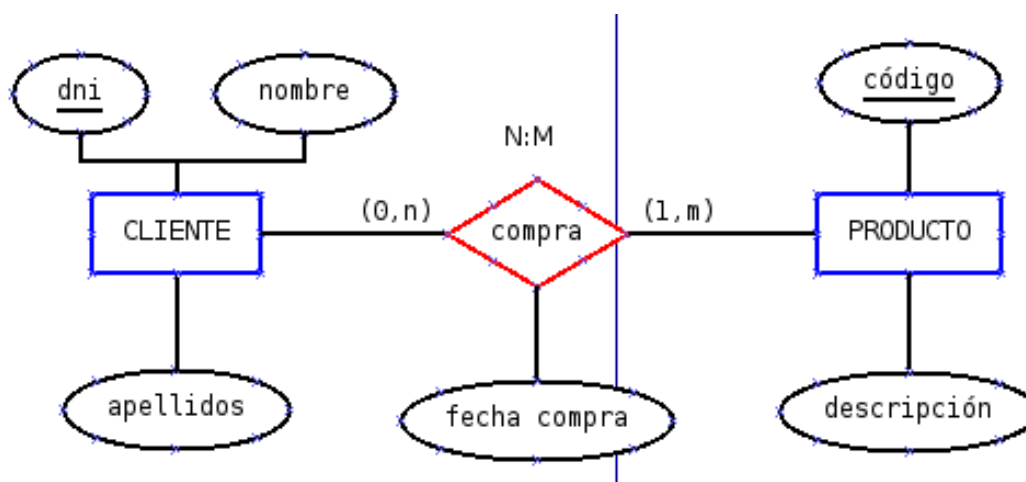
Toda relación N:M se transforma en una tabla que tendrá como clave primaria la concatenación de los atributos que asocia .



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)

2.3.2.- Transformación de relaciones 1-N.

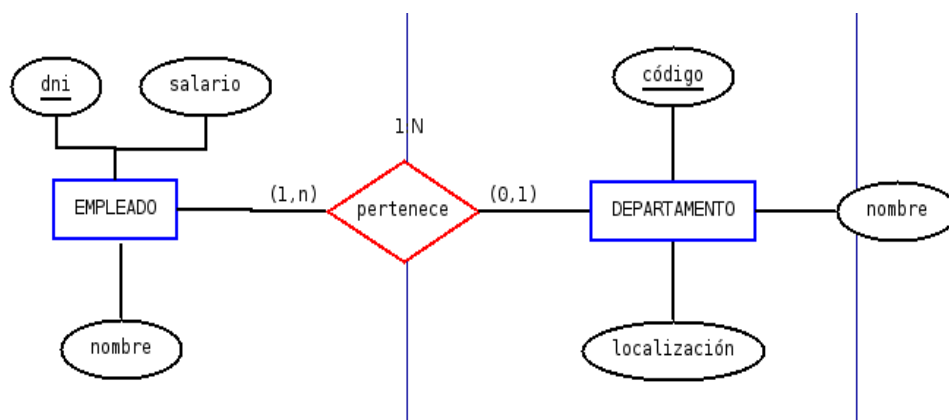
Existen dos soluciones:

- **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.

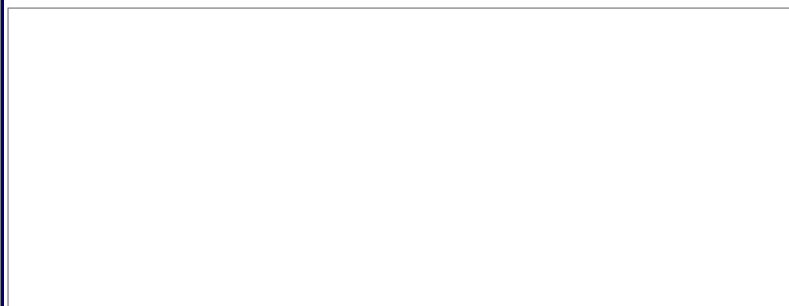
Así pues, se crea una nueva tabla formada por dni de EMPLEADO y código de DEPARTAMENTO. En esta nueva tabla dni de EMPLEADO será la clave primaria. El **modelo relacional** quedaría de la siguiente forma:

- 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.

Ejemplo1: Vamos a pasar el siguiente Modelo E/R al Modelo Relacional.



Tenemos dos entidades DEPARTAMENTO y EMPLEADO.

- Tabla DEPARTAMENTO: NumDepart, Nombre, Localidad.
- Tabla EMPLEADO: NumEmple, Apellido, Salario, Comision.

DEPARTAMENTO (NumDepart, Nombre, Localidad)

EMPLEADO (NumEmple, Apellido, Salario, Comision)

- Id Tabla DEPARTAMENTO: NumDepart (No puede haber dos departamentos con la misma identificacion).

- Id Tabla EMPLEADOS: NumEmple (No puede haber dos empleados con la misma identificacion)

DEPARTAMENTO (NumDepart, Nombre, Localidad)

EMPLEADO (NumEmple, Apellido, Salario, Comision)

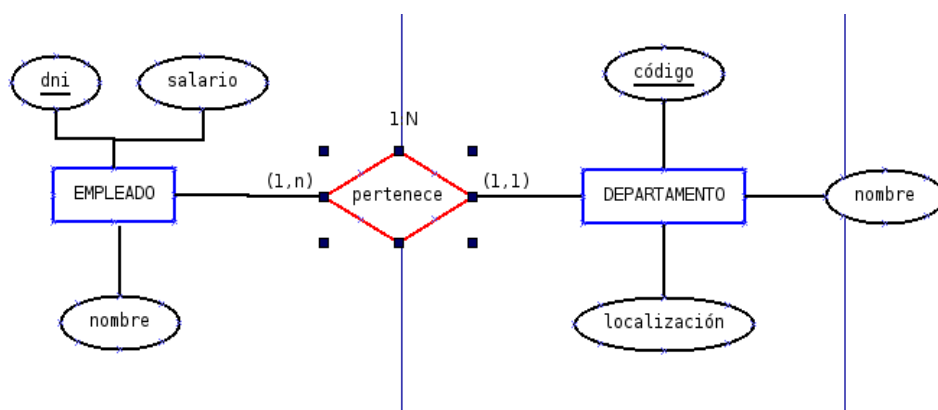
- En la relación ‘Pertenece’, se propaga el atributo de la entidad que tiene 1 como cardinalidad máxima (DEPARTAMENTO) hacia la que tiene cardinalidad máxima N (EMPLEADO). Se propaga NumDepart

(la clave). La Tabla EMPLEADOS tiene ahora esta estructura:

EMPLEADO (NumEmple, Apellido, Salario, Comision, NumDepart)

Atención: El ejemplo no está completo. Aún nos queda por tratar lo que hacemos con la relación “JEFE”. de cómo tratamos la relación recursiva “JEFE”, hablaremos en el punto 2.3.4.

Ejemplo2: 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)

2.3.3.- 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:

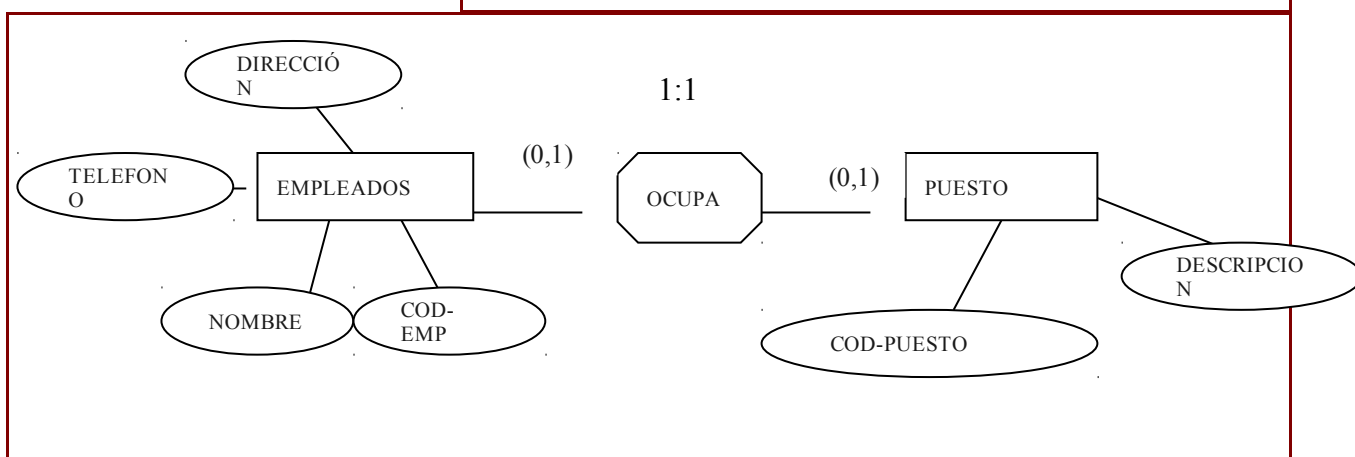
Supongamos que disponemos de una entidad EMPLEADOS, que contiene atributos relativos a los empleados de una empresa.

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:

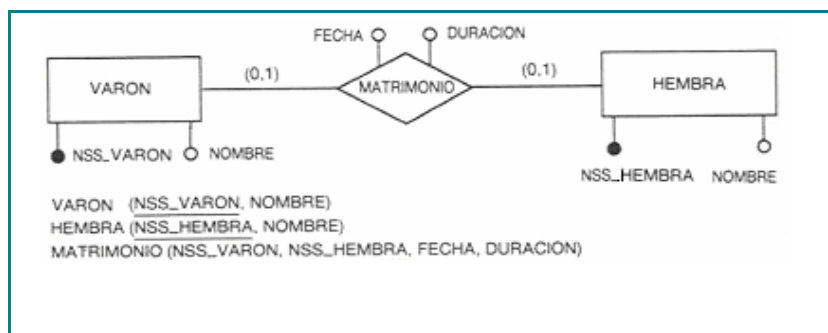


EMPLEADOS: (cod_empleado, nombre, dirección, teléfono).

PUESTO: (cod_puesto, descripción).

OCUPA: (cod_empleado, cod_puesto).

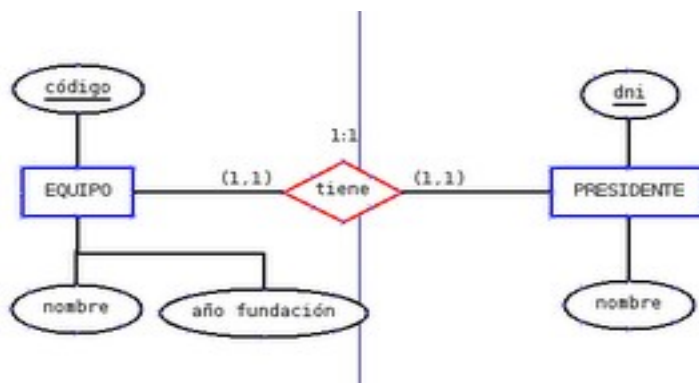
Ejemplo2:



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)

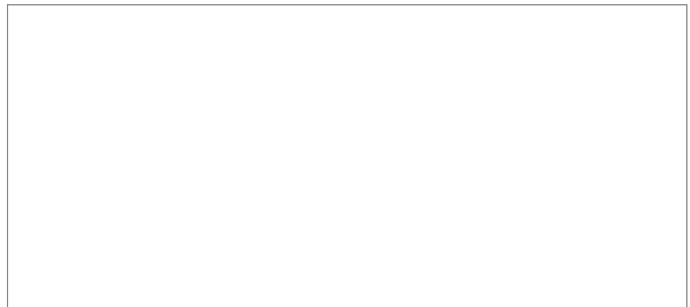
2.3.4.- Transformación de relaciones reflexivas o recursivas.

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: Veamos el modelo E-R que analizamos en el punto 2.3.2. (relaciones 1-N). Vemos que existe una relación “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 (NumEmple, Apellido, Salario, Comision, NumDepart, 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.

2.3.5.- 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 genera 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.

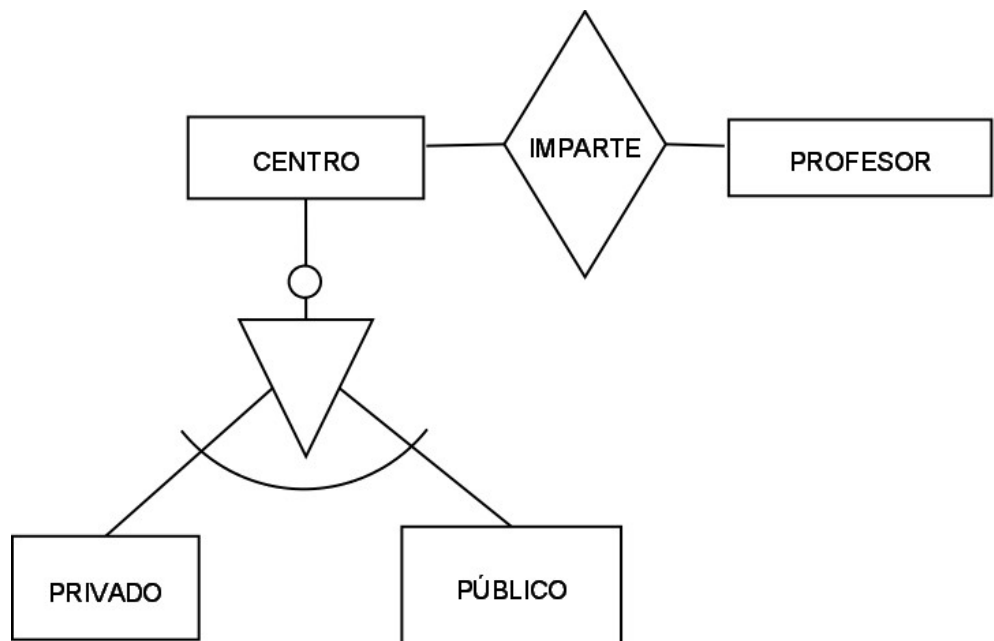
Consideramos la asignatura como atributo de la relación entre profesores y centro profesores –imparten - clase

Los centros educativos solo 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.

Modelo ER⁶:



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

Imparte: (códigos, asignatura)

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

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