

# INTRODUCCIÓN

- **BASE DE DATOS o B.D. (o D.B. o Data Base):** Colección de datos relacionados entre sí, estructurados (mediante campos) y organizados (ordenados).
- **SISTEMA GESTOR DE BASES DE DATOS o S.G.B.D. (o D.B.M.S. o Data Base Management System):** Un conjunto de programas que acceden y gestionan los datos de una base de datos.
- **Incoherencia o inconsistencia:** En una BD, existe inconsistencia en sus datos cuando diversas copias del mismo dato no concuerdan entre sí.
- **El subsistema de integridad:** En un SGBD, este mecanismo se encarga de detectar y corregir en la medida de lo posible operaciones incorrectas que pueden crear inconsistencias. Vigila la integridad de la BD.
- **Reglas de validación de un campo:** Expresiones que limitan los valores que pueden almacenarse en un campo. Por ejemplo, en Access, un campo numérico que únicamente puede tomar valores de 1 a 99, podría tener la regla de validación siguiente: ">0 Y <100".
- **Máscara de entrada de un campo:** Formato general o patrón físico que deben respetar los datos que se inserten en ese campo. Por ejemplo, en Access, un código postal podría ponerse como un campo de texto de 5 caracteres con la siguiente máscara de entrada "00000;\_;" (5 cifras numéricas obligadas con un carácter de subrayado indicando los huecos que aún quedan por rellenar).

El objetivo primordial del SGBD es proporcionar eficiencia y seguridad a la hora de extraer o almacenar información en la BD. Permiten gestionar grandes bloques de información, incluyendo las estructuras de los datos para su almacenamiento y los mecanismos para su gestión.

Una BD es un gran almacén de datos que pueden ser accedidos simultáneamente por varios usuarios, están relacionados y existe un número mínimo de duplicidad.

Los metadatos son las descripciones de los datos (tipos de los campos, tamaños, etc...) que se almacenan en la base de datos, y se encuentran en el diccionario de datos.

El SGBD es una aplicación que permite a los usuarios definir, crear y mantener la BD, y proporciona un acceso controlado a la misma. Debe prestar los siguientes servicios:

- **Creación y definición de la BD.** El SGBD permite la especificación de la estructura, el tipo de los datos, las restricciones y las relaciones entre ellos mediante "*lenguajes de definición de datos*". Toda esta información se almacena en el diccionario de datos.
- **Manipulación de los datos.** El SGBD permite realizar consultas sobre los datos almacenados, inserciones de nuevos datos, y actualizaciones sobre los datos ya existentes, utilizando "*lenguajes de manipulación de datos*".
- **Acceso controlado a los datos de la BD.** El SGBD permite mantener la seguridad para controlar el acceso no autorizado a los datos por usuarios sin permisos establecidos.
- **Mantenimiento de la integridad y consistencia.** El SGBD permite mantener la coherencia de los datos a través de mecanismos correctores que controlan los cambios no autorizados en los datos.
- **Acceso compartido a la BD.** El SGBD permite la interacción entre usuarios concurrentes.
- **Mecanismos de copias de seguridad y recuperación de los datos.** El SGBD permite, tras un fallo del sistema que ha provocado errores en los datos, recuperar los datos desde algún sistema de almacenamiento externo donde se realizó una copia de seguridad recientemente.

# COMPONENTES DE LOS SGBD:

## Lenguajes de los SGBD:

Todos los SGBD ofrecen lenguajes e interfaces apropiadas para cada tipo de usuario: administradores, diseñadores, programadores de aplicaciones que acceden a la BD y usuarios finales. Dichos lenguajes se clasifican en:

1. **Lenguaje de Definición de Datos (LDD o DDL):** Permiten especificar el esquema conceptual e interno de la BD, es decir, crea las entidades, describe sus atributos, etc... Es utilizado por diseñadores y administradores de la BD.
2. **Lenguaje de Manipulación de Datos (LMD o DML):** Permiten leer y actualizar los datos de la BD. Los objetos creados en la BD con un lenguaje de definición de datos se gestionan con un lenguaje de manipulación de datos. Los usuarios lo utilizan para realizar consultas, inserciones, eliminaciones y modificaciones sobre los datos de la BD. Las BD relacionales utilizan lenguajes como SQL (Structured Query Language) o QBE (Query By Example).
3. **Lenguaje de control de datos (LCD o DCL):** Permiten conceder o suprimir privilegios a los usuarios, es decir, realiza el control del acceso a los datos. Con este lenguaje se establecen las vistas de los usuarios, así a cada usuario se le permite manipular únicamente el conjunto de datos que le interesan, y se le deniega el acceso a los datos que no necesita.
4. **Lenguajes de control de transacciones:** Controlan los cambios realizados en los datos de la BD mediante instrucciones de manipulación de datos (DML). Permiten agrupar varias instrucciones DML como si fuesen una única instrucción lógica de manipulación, de forma que o se realiza el grupo de instrucciones completo o se deshacen los cambios de todo el grupo (esto ocurriría en el caso de que simplemente una de las instrucciones DML del grupo no pudiera realizarse o generara algún error en su ejecución).
5. **Lenguajes de cuarta generación (4GL) o Herramientas de Desarrollo:** La mayoría de los SGBD comerciales los incluyen, y permiten al usuario crear aplicaciones de forma fácil y rápida para acceder y manipular los datos de la BD.
  - Por ejemplo, SQL Forms de Oracle, que permite crear formularios para interactuar con los datos; SQL Reports de Oracle, para generar informes de los datos contenidos en la BD; PL/SQL de Oracle, que es un lenguaje que permite crear procedimientos que interactúen con los datos de la BD.

## El Diccionario de Datos:

Es el lugar donde se guarda información acerca de todos los datos que forman la BD: su descripción y la de los objetos que la forman. En una BD relacional, el diccionario de datos proporciona información acerca de:

- La estructura lógica y física de la BD. Esquemas externo, conceptual e interno, y correspondencia entre los esquemas.
  - Las definiciones de todos los objetos de la BD: tablas, vistas, índices, disparadores, procedimientos, funciones, etc.
  - El espacio asignado y utilizado por los objetos.
- Los valores por defecto de las columnas de las tablas.
- Información acerca de las restricciones de integridad.
- Los privilegios y roles otorgados a los usuarios.
- Estadísticas de utilización, tales como la frecuencia de las transacciones y el número de accesos realizados a los objetos de la base de datos.
- Se puede tener un historial de los cambios realizados sobre la base de datos.

## Seguridad e Integridad de los Datos:

Un SGBD proporciona los siguientes mecanismos para garantizar la seguridad e integridad de los datos:

---

- Debe garantizar la protección de los datos contra accesos no autorizados, tanto intencionados como accidentales.
- Debe implantar restricciones de integridad que protegerán la BD contra daños accidentales, pues los valores de los datos que se quieren almacenar deberán satisfacer ciertos tipos de restricciones de consistencia y reglas de integridad, especificadas por el administrador de la BD. El SGBD puede determinar si se produce una violación de la restricción impuesta.
- Debe proporcionar herramientas y mecanismos para la planificación y realización de copias de seguridad, y su posible posterior restauración tras un fallo del sistema.
- Debe ser capaz de recuperar la BD llevándola a un estado consistente en caso de ocurrir algún suceso que la dañe.
- Debe asegurar el acceso concurrente y ofrecer mecanismos para conservar la consistencia de los datos en el caso de que varios usuarios actualicen la BD de forma concurrente.

### El Administrador de la BD:

En los SGBD existen distintos tipos de usuarios, cada tipo con unos permisos o privilegios diferentes sobre los objetos que forman la BD.

Por ejemplo, en los sistemas Oracle los tipos de usuarios más importantes son:

1. Los usuarios de la categoría **DBA (Database Administrador)**: Con el nivel más alto de privilegios, pues son los administradores de la base de datos.
2. Los usuarios de la categoría **RESOURCE**: Pueden crear sus propios objetos y únicamente tienen acceso a los objetos para los que se les ha dado permiso.
3. Los usuarios del tipo **CONNECT**: Únicamente pueden tener acceso a los objetos para los que se les ha concedido permiso.

El DBA tiene una gran responsabilidad al tener el máximo nivel de privilegios. Hay que intentar que haya solo uno o muy pocos. Los DBA crearán los demás usuarios y les asignarán sus tipos y permisos de acceso.

Entre las tareas de un DBA están:

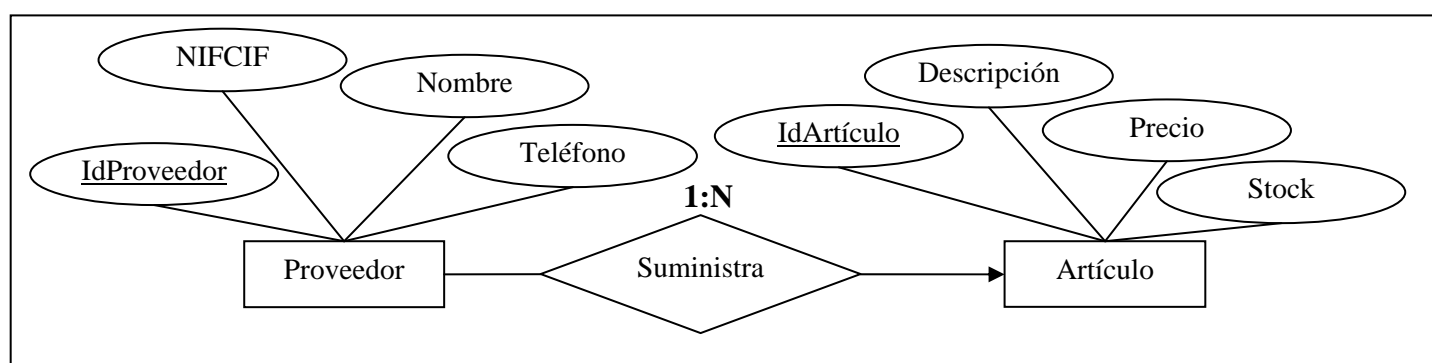
1. Instalar el SGBD en el sistema informático de la empresa.
2. Crear las BD que se vayan a gestionar.
3. Crear y mantener el esquema de la BD.
4. Crear y mantener las cuentas de usuario de la BD.
5. Arrancar y parar el SGBD, y cargar las BD con las que se ha de trabajar.
6. Colaborar con el Administrador del SO en las tareas de ubicación, dimensionado y control de los archivos y espacios de disco ocupados por el SGBD.
7. Colaborar en las tareas de formación de los usuarios.
8. Establecer estándares de uso, políticas de acceso y protocolos de trabajo diario para los usuarios de la BD.
9. Suministrar la información necesaria sobre la BD a los equipos de análisis y programación de aplicaciones.
10. Efectuar tareas de explotación como:
  - 10.1. Vigilar el trabajo diario colaborando en la información y resolución de las dudas de los usuarios.
  - 10.2. Controlar en tiempo real los accesos, tasas de uso, cargas en los servidores, anomalías, etc.
  - 10.3. Llegado el caso, reorganizar la BD.
  - 10.4. Efectuar las copias de seguridad periódicas de la BD.
  - 10.5. Restaurar la BD después de un incidente material a partir de las copias de seguridad.
  - 10.6. Estudiar las auditorías del sistema para detectar anomalías, intentos de violación de la seguridad, etc.
  - 10.7. Ajustar y optimizar la BD mediante el ajuste de sus parámetros, con ayuda de las herramientas de monitorización y de las estadísticas del sistema.

## Conceptos previos:

- ✓ Los **modelos de datos** son el conjunto de **conceptos o herramientas conceptuales** que **sirven para describir la estructura de una BD**: los datos, las relaciones y las restricciones que se deben cumplir sobre los datos.
- ✓ Se denomina **esquema de la BD** a la **descripción de una BD mediante un modelo de datos**. Este esquema se especifica durante el diseño de la BD.

# EL MODELO DE DATOS ENTIDAD-RELACIÓN:

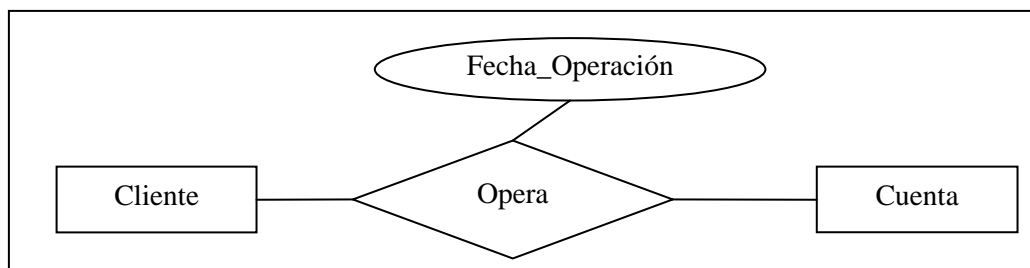
- El Modelo Entidad-Relación (o Modelo E-R o Modelo Entidad-Interrelación) fue propuesto por Peter Chen en 1976 para la representación conceptual de los problemas del mundo real. Este modelo de datos representa los datos utilizando grafos y símbolos gráficos, además de tablas para la representación de los datos y sus relaciones.



## Conceptos básicos usados en el Modelo E-R:

1. **Entidad:** Es un objeto del mundo real que tiene interés para la empresa. Por ejemplo, la entidad ALUMNO de un centro escolar o la entidad CLIENTE de una empresa. Se representan con rectángulos con el nombre en el interior.
2. **Conjunto de Entidades:** Es un grupo de entidades del mismo tipo, y no tienen que ser conjuntos disjuntos, es decir, puede haber una entidad que pertenezca a varios conjuntos de entidades a la vez. Por ejemplo, el conjunto de entidades ALUMNOS de un centro escolar.
3. **Entidad Fuerte:** Es una entidad que no depende de otra entidad para su existencia. Por ejemplo, la entidad ALUMNO es fuerte pues no depende de otra para existir como entidad, mientras que la entidad NOTA es una entidad débil pues necesita a la entidad ALUMNO para existir.
4. **Atributos o Campos:** Son las unidades de información que describen propiedades de las entidades. Por ejemplo, la entidad ALUMNO posee los atributos: número de matrícula, nombre, dirección, población, código postal, provincia, y teléfono. Los atributos toman valores, por ejemplo, el atributo provincia podría ser SEVILLA, CÁDIZ, etc. Se representan mediante una elipse con el nombre en el interior.
5. **Dominio:** Es el conjunto de valores permitidos para cada atributo. Por ejemplo, el dominio del atributo nombre puede ser el conjunto de cadenas de texto de una longitud determinada.
6. **Identificador o Superclave:** Es el conjunto de atributos que identifican de forma única a cada entidad. Por ejemplo, la entidad EMPLEADO, con los atributos: número de la seguridad social, dni, nombre, dirección, fecha de nacimiento y teléfono, podría tener como identificador sólo el dni (pues no habrá 2 empleados con el mismo dni), o sólo el número de la seguridad social, o el conjunto de 3 atributos nombre, fecha de nacimiento y teléfono (pues es difícil que hay 2 empleados en la misma empresa que tengan los mismos valores en esos 3 atributos).

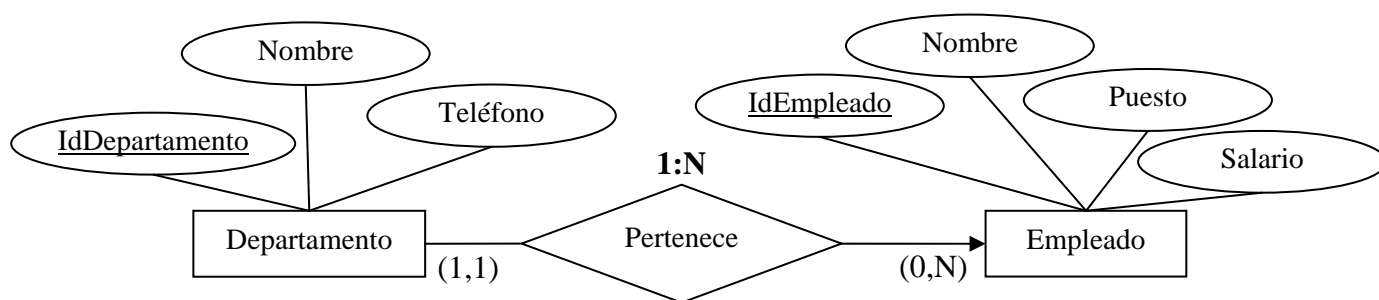
7. **Clave Candidata:** Es cada una de las superclaves formadas por el mínimo número de campos posibles. En el ejemplo anterior habría 2 claves candidatas de un único atributo: dni o número de la seguridad social.
8. **Clave Primaria o Clave Principal (Primary Key):** Es la clave candidata seleccionada por el diseñador de la BD para identificar a cada entidad. Una clave primaria no puede tener valores nulos (vacíos), ha de ser sencilla de crear y no ha de variar con el tiempo. El atributo o conjunto de atributos que forman parte de la clave primaria se representan subrayados.
9. **Clave Ajena o Clave Foránea (Foreign Key):** Es el atributo o conjunto de atributos de una entidad que constituyen la clave primaria de otra entidad. Las claves foráneas representan las relaciones entre entidades. Por ejemplo, la entidad ARTÍCULO con los atributos: código de artículo, descripción de artículo, precio de venta y stock en almacén, y la entidad VENTA con los atributos: código de venta, fecha de venta, código de artículo y unidades vendidas; pues el atributo código de artículo es clave foránea en la entidad VENTA, pues la relaciona con la entidad ARTÍCULO, debido a que ese atributo es clave primaria de la entidad ARTÍCULO.
10. **Relación:** Es una asociación entre diferentes entidades. Se representan mediante un rombo con su nombre, un verbo, en su interior.
11. **Conjunto de Relaciones:** Es un grupo de relaciones del mismo tipo. Por ejemplo, entre los conjuntos de entidades ARTÍCULOS y VENTAS puede haber varias relaciones distintas, pues todas ellas pueden formar un conjunto de relaciones, que vinculan el conjunto de entidades ARTÍCULOS con el de VENTAS.
  - Una relación puede tener atributos descriptivos, por ejemplo, supongamos que la entidad CLIENTE está relacionada con la entidad CUENTA a través de una relación OPERA; se necesitaría el atributo FECHA\_OPERACIÓN en el conjunto de relaciones CLIENTE\_CUENTA, que especificaría la última fecha en la que el cliente tuvo acceso a su cuenta bancaria.



## Diagramas de estructuras de datos en el modelo E-R:

Los diagramas E-R representan la estructura lógica de una BD de manera gráfica. Los símbolos utilizados son:

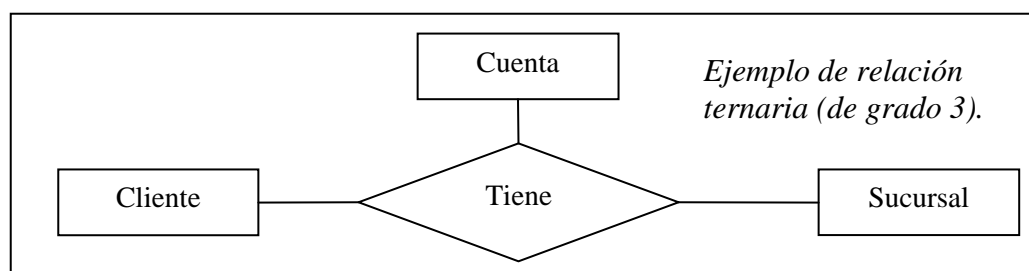
1. Rectángulos para representar entidades.
2. Elipses para los atributos.
3. Rombos para las relaciones.
4. Cada atributo se unirá a la entidad o a la relación a la que pertenezca con líneas simples.
5. Las líneas podrán tener forma de flecha en una relación. Donde esté la punta de la flecha estará el MUCHOS (N), y donde no hay punta de flecha en la línea estará el UNO (1). La orientación de la flecha señala la cardinalidad de la relación.
6. Cada componente gráfico se etiqueta con el nombre que lo representa.



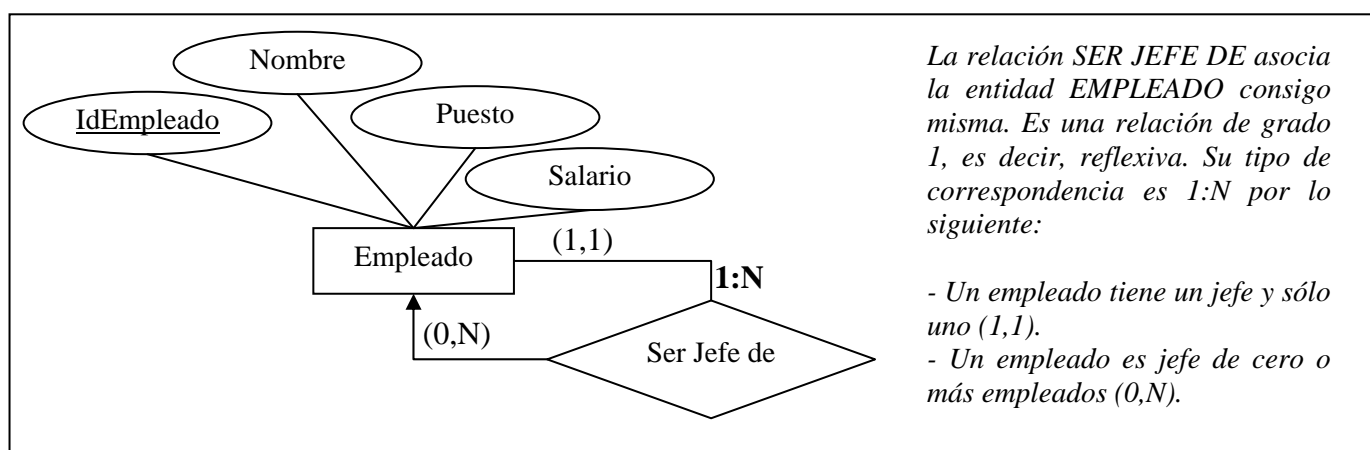
*Diagrama Entidad-Relación de las relaciones entre departamentos y empleados de una empresa. A cada departamento le pertenecen 0 o más empleados (N). Cada empleado pertenece a un departamento y sólo a uno.*

## Grado y cardinalidad de las relaciones:

El grado de una relación es el número de conjuntos de entidades que participan en el conjunto de relaciones, es decir, el número de entidades que participan en una relación. Lo normal es que las relaciones sean binarias (relaciones de grado 2), es decir, que en las relaciones participen 2 entidades. No obstante, puede haber relaciones ternarias (de grado 3) o incluso de otro grado, aunque son poco comunes. Las relaciones en las que sólo participa una entidad se llaman anillo o de grado 1 o relaciones reflexivas.



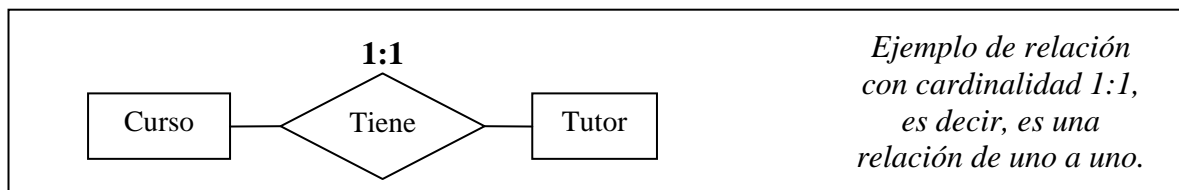
Un ejemplo de relación de anillo sería el siguiente: la entidad EMPLEADO puede tener una relación SER JEFE DE consigo misma, pues un empleado es jefe de muchos empleados y, a la vez, el jefe es un empleado. Otro ejemplo sería la relación SER DELEGADO DE los alumnos de un curso, pues el delegado es también alumno del curso.



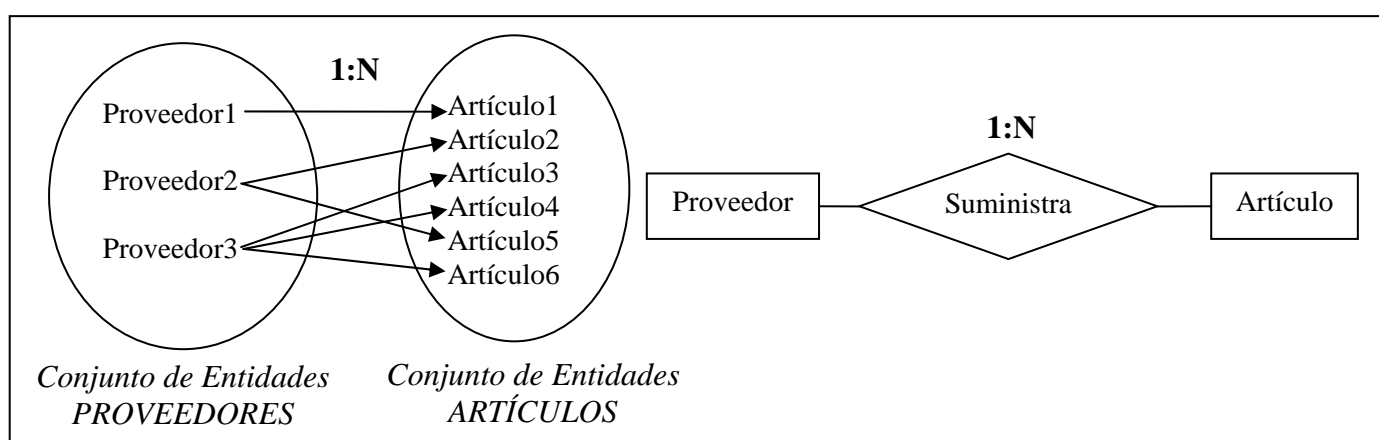
Las cardinalidades de asignación expresan el número de entidades a las que puede asociarse otra entidad mediante un conjunto relación. Las cardinalidades de asignación se describen únicamente para conjuntos binarios de relaciones.

Las cardinalidades de asignación son las siguientes:

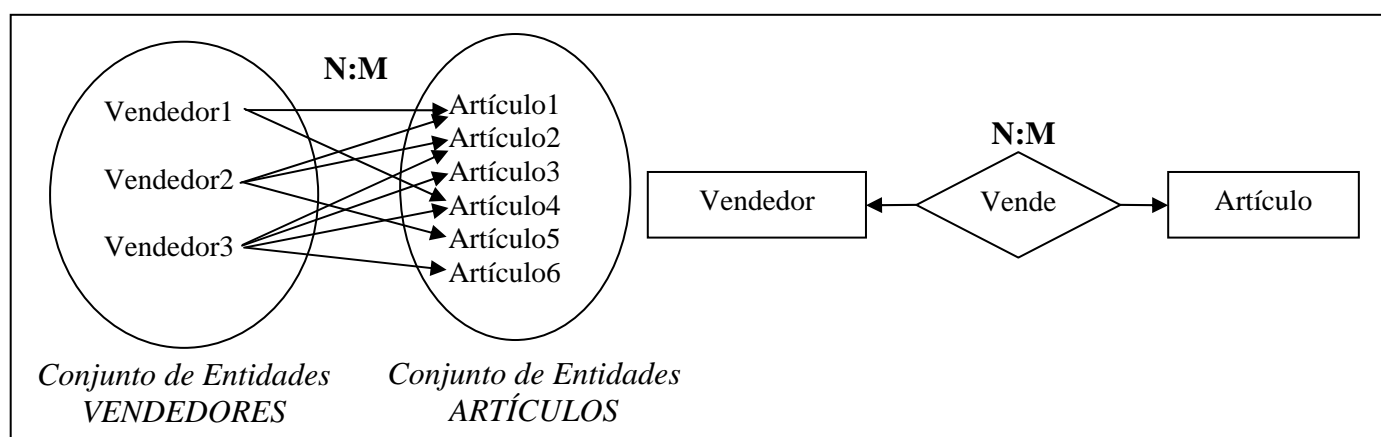
1. **1:1 o uno a uno:** A cada elemento de la primera entidad le corresponde sólo uno de la segunda entidad, y a la inversa. *Por ejemplo, un cliente de un hotel ocupa una habitación y cada habitación es ocupada por un cliente titular, o por ejemplo, cada curso de alumnos tiene un único tutor, y ese tutor es únicamente tutor de ese curso.*



2. **1:N o uno a muchos:** 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 sólo de la primera entidad. *Por ejemplo, un mismo proveedor suministra varios artículos a una empresa, y cada artículo que adquiere la empresa siempre es pedido al mismo proveedor.*

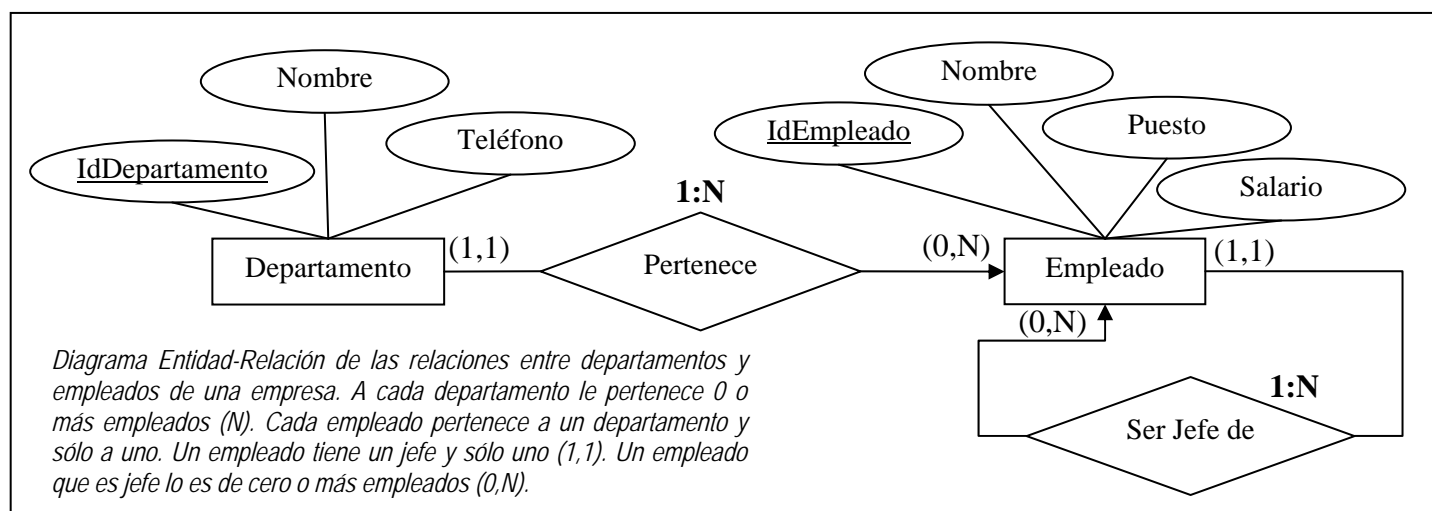


3. **N:M o muchos a muchos:** A cada elemento de la primera entidad le corresponde uno o más elementos de la segunda entidad, y a cada elemento de la segunda entidad le corresponde uno o más elementos de la primera entidad. *Por ejemplo, cada vendedor de una tienda vende muchos artículos y cada artículo es vendido por varios vendedores.*



La cardinalidad de una entidad informa del grado de participación de dicha entidad concreta en la relación. Se expresan entre paréntesis indicando los valores máximo y mínimo. Los valores son: (0,1), (1,1), (0,N), (1,N) y (N,M). El valor 0 se pone cuando la participación de la entidad es opcional.

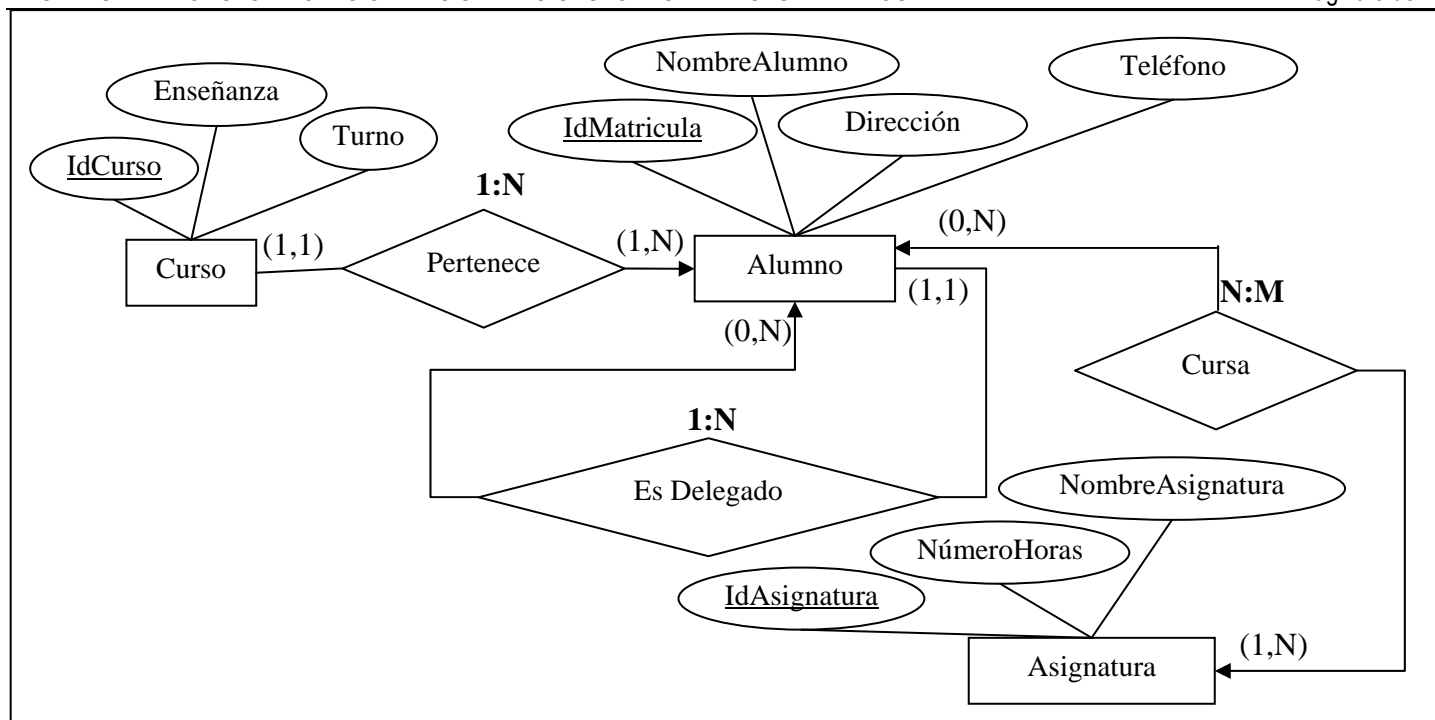
El ejemplo completo del diagrama E-R en el que se relacionaban las entidades EMPLEADO y DEPARTAMENTO sería:



**Ejemplo de diagrama de estructuras de datos en el modelo E-R:** En un centro escolar se imparten muchos cursos. Y cada curso está formado por un grupo de alumnos, dentro de cada curso un alumno es el delegado del grupo. Los alumnos cursan asignaturas, y una asignatura puede o no ser cursada por los alumnos.

- Se identifican las entidades (mirando los nombres del enunciado): ~~centro~~ (es un único centro, el nuestro), ~~curso~~, ~~alumno~~, ~~delegado~~ (pues son alumnos), ~~asignatura~~.
- Se identifican las relaciones (mirando los verbos que unen entidades interesantes): cada curso formado por alumnos, un alumno es delegado, alumnos cursan asignaturas.
- Se identifican las cardinalidades de las relaciones (examinando detenidamente el enunciado).
- Se identifican los atributos de cada entidad: puesto que el enunciado no informa de los posibles atributos de cada entidad tendremos que suponerlos:
  - Curso (IdCurso, Enseñanza, Turno).
  - Alumno (IdMatricula, NombreAlumno, Dirección, Teléfono).
  - Asignatura (IdAsignatura, NombreAsignatura, NúmeroHoras).
- Deducciones:
  - Un curso está formado por muchos alumnos, y cada alumno pertenece a un curso, por tanto la relación binaria es 1:N. Para calcular la cardinalidad de las entidades alumno y curso preguntamos: A un curso ¿cuántos alumnos pertenecen como mínimo? ¿y como máximo? Al menos debe haber un alumno para que haya curso, y como máximo muchos, por tanto la cardinalidad de la entidad alumno es (1,N). Un alumno ¿a cuántos cursos va a pertenecer? Como mínimo a 1 y como máximo también a 1, por tanto, la cardinalidad de la entidad curso es (1,1).
  - De los alumnos que hay en un grupo uno de ellos es delegado, es una relación de grado 1. La relación es del tipo 1:N porque un alumno es delegado de muchos alumnos. ¿Un alumno de cuántos alumnos es delegado? Como mínimo de 0 pues puede que ese alumno no sea el delegado, y como máximo de muchos, pues si es el delegado lo será de todos los demás alumnos del grupo, por tanto, la cardinalidad de la entidad alumno en ese extremo es (0,N); y en el otro, como cada alumno tiene un delegado como mínimo y como máximo, pondremos (1,1).
  - Entre alumnos y asignaturas surge una relación binaria N:M, pues un alumno cursa varias asignaturas y una asignatura es cursada por varios alumnos. La cardinalidad de la entidad alumno en la relación cursa será (1,N), pues un alumno como mínimo cursa una asignatura y como máximo varias, y la cardinalidad de la entidad asignatura para esa relación será (0,N) pues puede que exista una asignatura a la que no se haya matriculado ningún alumno, por tanto cursada por 0 alumnos como mínimo, o por muchos como máximo.





## Generalización y Jerarquías de Generalización:

Las generalizaciones proporcionan un mecanismo de abstracción que permite especializar una entidad, denominada supertipo, en subtipos. También se dice que se generalizan 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 identificarán unas características. Los atributos comunes describirán el supertipo y los particulares los subtipos.

Por ejemplo, en una empresa de construcción se identifican las siguientes entidades:

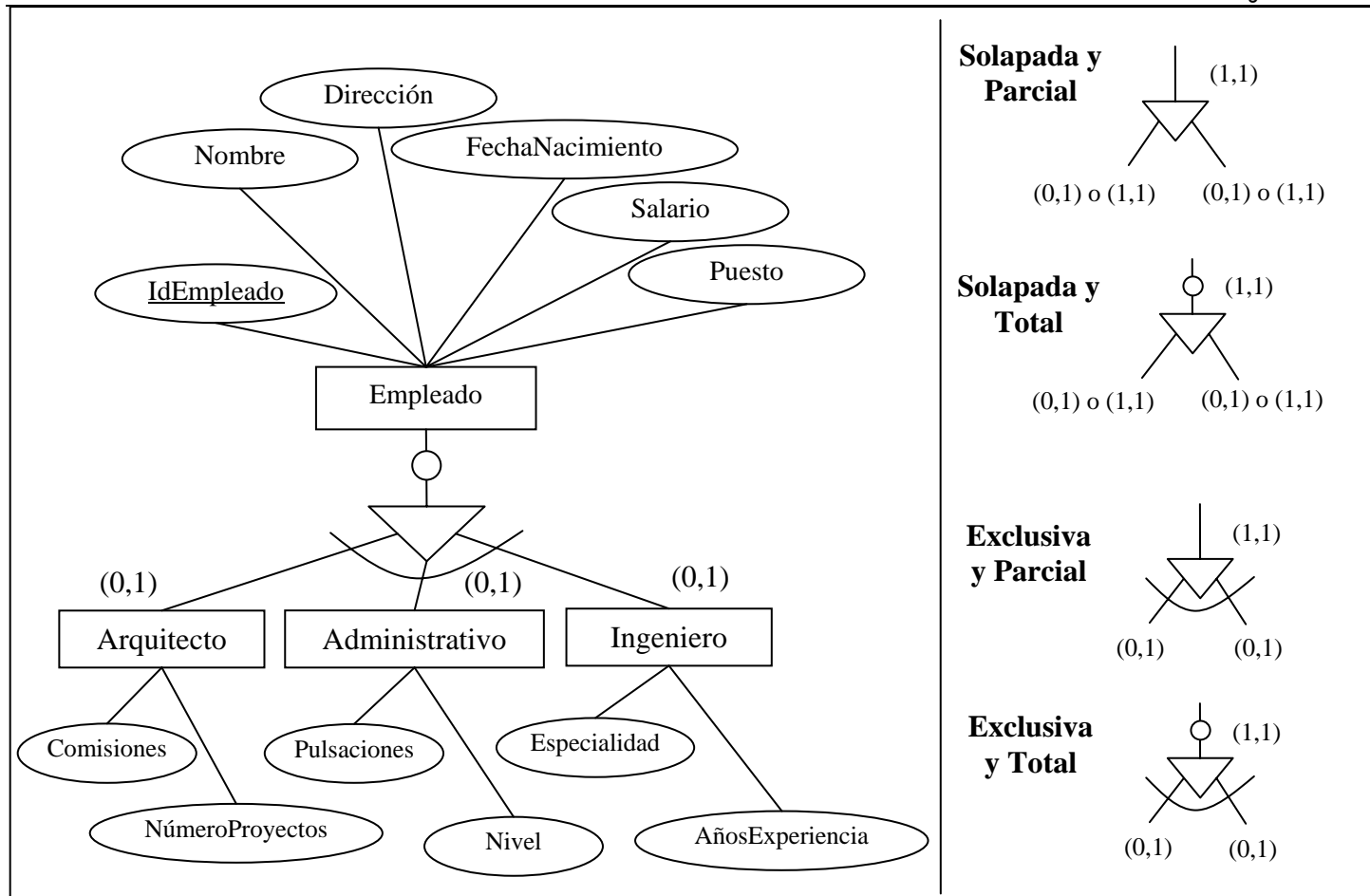
- **Empleado** (IdEmpleado, Nombre, Dirección, FechaNacimiento, Salario, Puesto)
  - **Arquitecto**, que incluye los atributos de un empleado más los atributos específicos: NúmeroProyectos y Comisiones.
  - **Administrativo**, que incluye los atributos de un empleado más los atributos específicos: Pulsaciones y Departamento.
  - **Ingeniero**, que incluye los atributos de un empleado más los atributos específicos: Especialidad y AñosExperiencia.

La herencia es el mecanismo por el que los atributos del supertipo sean “heredados” por sus subtipos.

La generalización es total si no hay ocurrencias en el supertipo que no pertenezcan a ninguno de sus subtipos, es decir, los empleados de la empresa, o son arquitectos, o administrativos, o ingenieros, no hay de otro tipo.

La generalización es parcial si hay empleados que no pertenecen a ningún subtipo, es decir, que exista algún empleado que no sea ni arquitecto, ni administrativo, ni ingeniero.

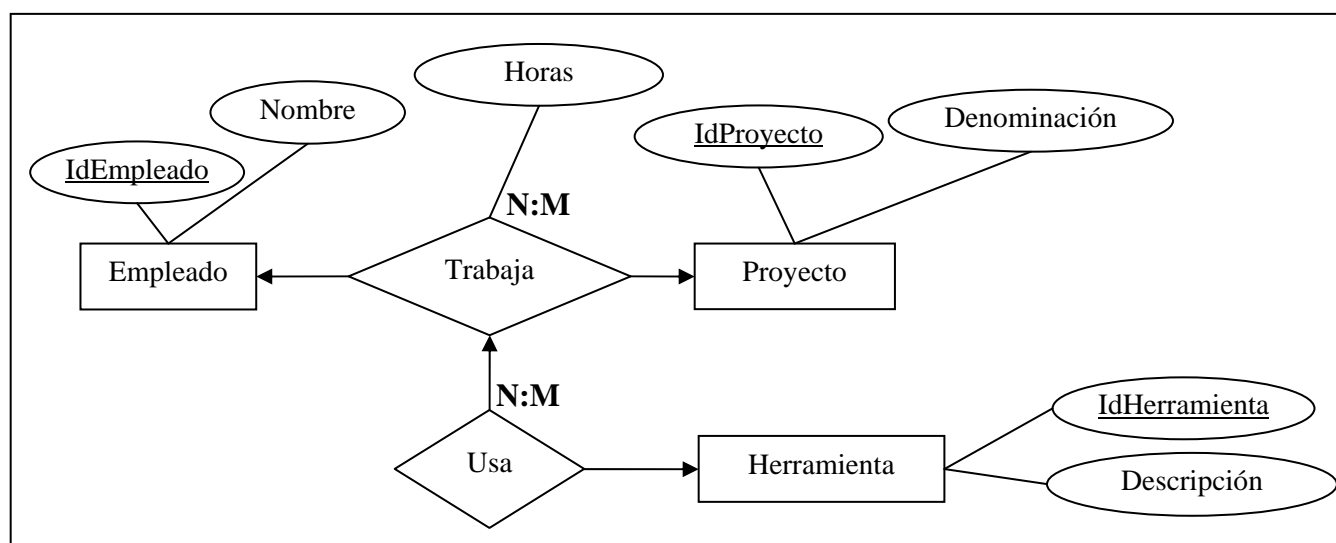
La generalización es exclusiva si cada empleado sólo puede pertenecer a un único subtipo, y no a más de uno. Si un empleado puede ser varias cosas a la vez la generalización es solapada o superpuesta.



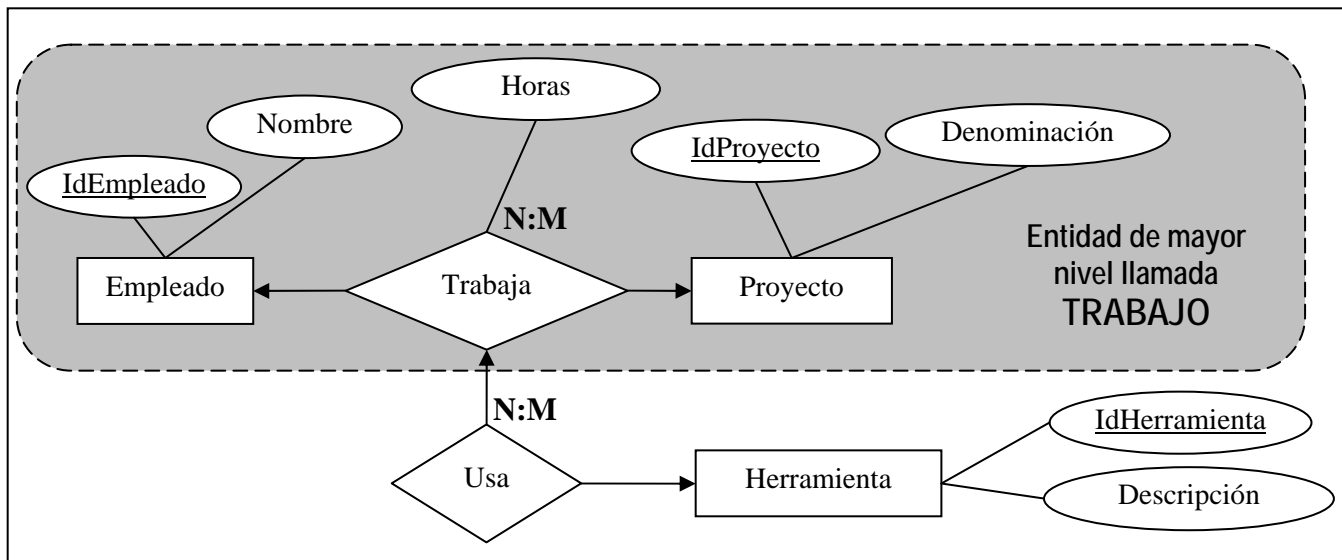
## Agregación:

Una limitación del modelo E-R es que *no es posible expresar relaciones entre relaciones*. En estos casos se realiza una agregación, que es una abstracción a través de la cual las relaciones se tratan como entidades de más alto nivel.

Por ejemplo, un empleado *trabaja* una serie de “horas” asignadas, en varios proyectos, y *usa* unas herramientas determinadas durante la realización de los proyectos. La representación del diagrama E-R sería:

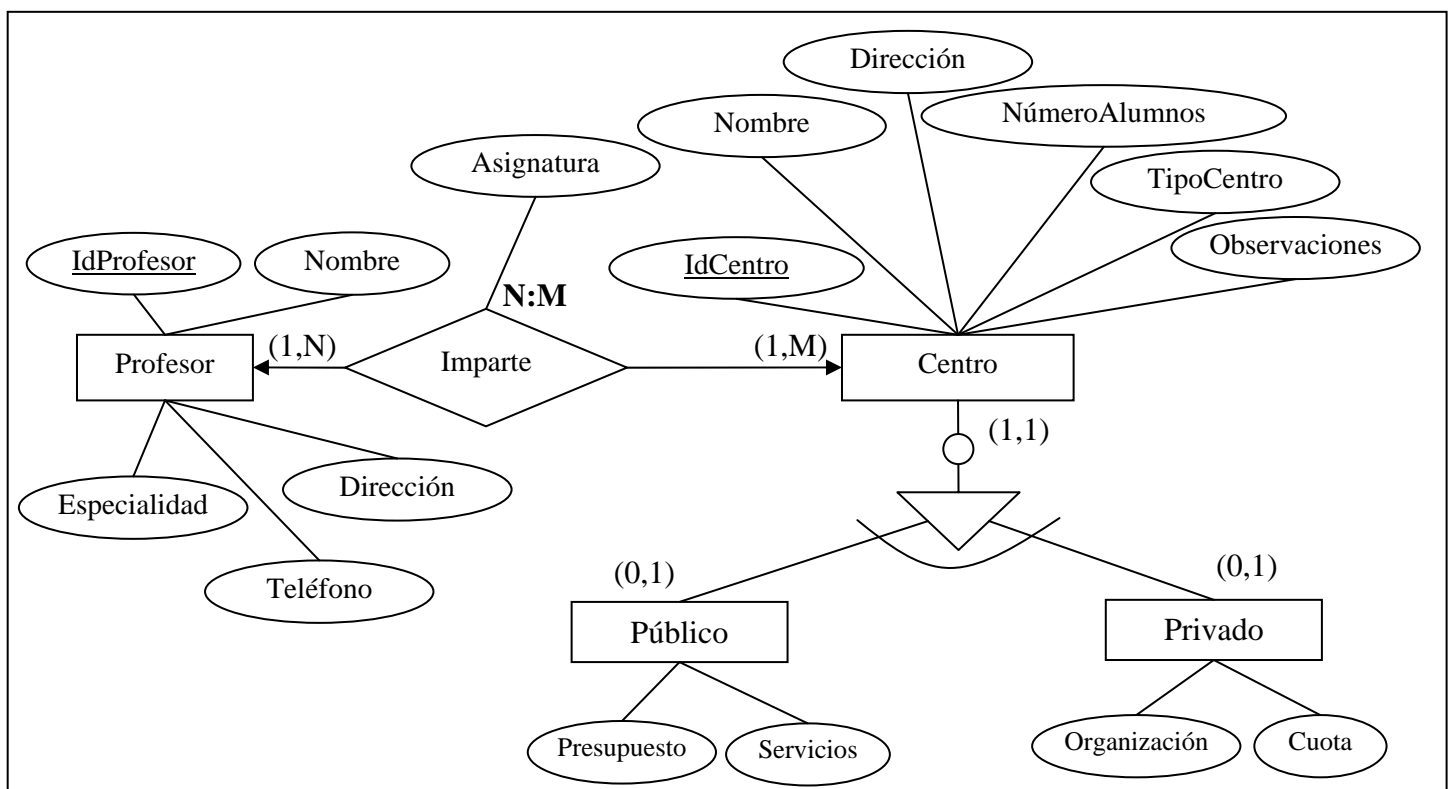


Teniendo en cuenta la agregación, se puede considerar que existe una entidad de mayor nivel llamada TRABAJO, compuesta por relación TRABAJO y las entidades EMPLEADO y PROYECTO. Esta entidad de mayor nivel es realmente un conjunto de entidades y relaciones que están asociadas. Al ser considerado el conjunto como una única entidad, ya puede utilizarse una relación, en este caso la relación USA, para asociarla a otra entidad. Así conseguimos relacionar una relación con una entidad (la relación TRABAJA con la entidad HERRAMIENTA a través de la relación USA).



- Ejercicio de diagrama E-R:

Hay profesores que *imparten* clases en 2 tipos de centros educativos: públicos y privados. Un profesor puede impartir clase en varios centros, ya sean públicos o privados. La asignatura será un atributo de la relación entre el profesor y el centro donde imparte. Los centros educativos sólo pueden ser públicos o privados. Un centro público no puede ser privado a la vez, ni a la inversa. Los atributos específicos para los centros públicos son: el presupuesto y los servicios; y para los privados son: la organización y la cuota.



# EL MODELO RELACIONAL

**NOTA:** Abreviaturas usadas: BD = Base de datos, SGBD = Sistema gestor de bases de datos, BDR = BD relacional, SGBDR = SGBD relacional.

## Conceptos previos vistos anteriormente:

- ✓ Los modelos de datos son el conjunto de conceptos o herramientas conceptuales que sirven para describir la estructura de una BD: los datos, las relaciones y las restricciones que se deben cumplir sobre los datos.
- ✓ Se denomina esquema de la BD a la descripción de una BD mediante un modelo de datos. Este esquema se especifica durante el diseño de la BD.
  - El Modelo Relacional, el más usado, representa los datos y sus relaciones mediante una colección de **tablas**, cuyas **columnas (campos)** tienen nombre únicos y hay una por cada atributo, y las **filas (tuplas)** representan a los **registros guardados en dicha tabla**.

El modelo relacional fue desarrollado por E. F. Codd para IBM a finales de los años sesenta, y mantiene la independencia de la estructura lógica de la base de datos respecto al modo de almacenamiento y otras características de tipo físico.

El modelo relacional persigue, al igual que la mayoría de los modelos de datos, los siguientes objetivos:

1. **Independencia física de los datos:** El modo de almacenamiento de los datos no debe influir en su manipulación lógica.
2. **Independencia lógica de los datos:** Los cambios que se realicen en los objetos de la base de datos *sin alterar los datos almacenados previamente* no deben repercutir en los programas y usuarios que acceden a la misma.
3. **Flexibilidad:** Para representar a los usuarios los datos de la forma más adecuada a la aplicación que utilicen.
4. **Uniformidad:** En la presentación de las estructuras lógicas de los datos, que son tablas, lo que facilita la concepción y manipulación de la base de datos por parte de los usuarios.
5. **Sencillez:** Pues las características anteriores, así como unos lenguajes de usuario sencillos, hacen que este modelo sea fácil de comprender y utilizar por el usuario.

Codd introduce el concepto de relación (tabla) como estructura básica del modelo. Todos los datos de una BD se representan en forma de relaciones cuyo contenido varía en el tiempo. El modelo relacional se basa en 2 ramas de las matemáticas: la teoría de conjuntos y la lógica de predicados.

Relación EMPLEADOS			
Número_Empleado	Nombre_Completo	Número_Departamento	Fecha_Alta
154621	Andrés Pérez Ruiz	15	10/08/2001
889701	Nuria Gálvez Pinar	8	06/05/2002

Ejemplo de una relación (tabla) con 4 campos (columnas o atributos) y 2 tuplas (filas o registros).

## ESTRUCTURA DEL MODELO DE DATOS RELACIONAL:

La relación es el elemento básico del modelo relacional, y se representa como una tabla, con su nombre y las columnas que representan los atributos. A las filas de la tabla se las llama **tuplas**, y contienen los valores que toman cada uno de los atributos para cada elemento de la relación. Una representación de una relación en forma de tabla podría ser:

NúmeroMatrícula	Nombre	Apellidos	Curso
1023	Felipe	Ruiz Sánchez	1A-ASI
1089	Ana	Fernández Soto	2A-ASI
2112	Alfonso	Sanz Solés	1C-ESO

Un **dominio** es el conjunto finito de valores homogéneos (del mismo tipo) y atómicos (indivisibles) que puede tomar cada campo. Los valores contenidos en una columna pertenecen a un dominio previamente definido. Todos los dominios tienen un nombre y un tipo de datos asociado. Existen 2 tipos de dominios:

1. **Dominios generales:** Sus valores están comprendidos entre un máximo y un mínimo. Por ejemplo, el campo Código Postal está formado por números naturales de 5 cifras, del 00000 al 99999.
2. **Dominios restringidos:** Sus valores pertenecen a un conjunto de valores específico, por tanto, no están comprendidos en un rango entre un máximo y un mínimo. Por ejemplo, el campo Sexo puede tomar los valores H o M.

Por ejemplo, en la relación ALUMNOS podemos considerar los siguientes dominios:

- Atributo NúmeroMatrícula con el dominio: conjunto de números naturales formados por 4 dígitos.
- Atributo Nombre con el dominio: conjunto de 15 caracteres.
- Atributo Apellidos con el dominio: conjunto de 20 caracteres.
- Atributo Curso con el dominio: conjunto de 8 caracteres.

A los dominios habituales se les puede poner un nombre.

NOMBRES: dominio formado por conjuntos de 40 caracteres.

SALARIOS: dominio formado por número reales de 4 cifras en la parte entera y 2 en la decimal.

Así puede describirse el campo Nombre como con el dominio NOMBRES.

Una relación (o tabla) está formada por:

1. **Atributo (o campo):** Cada columna de la tabla. Tiene su propio nombre y pueden guardar un conjunto de valores. El orden de las columnas en una tabla es irrelevante, una columna se identifica por su nombre no por su posición.
2. **Tupla (o registro):** Cada fila de la tabla.

De las tablas se derivan los siguientes conceptos:

1. **Cardinalidad:** Número de filas de la tabla. La relación ALUMNOS tenía cardinalidad 3.
2. **Grado:** Número de columnas de la tabla. La relación ALUMNOS tenía grado 4.
3. **Valor:** Intersección entre una fila y una columna. Un valor de la relación ALUMNOS puede ser 1089.

4. **Valor Null (o valor nulo):** Representa la ausencia de información.

#### Propiedades de las relaciones:

1. No puede haber 2 relaciones con el mismo nombre en la BD.
2. En cada tupla (fila), cada atributo toma un único valor (tienen valores atómicos). Se dice que las relaciones están normalizadas.
3. No puede haber 2 atributos con el mismo nombre en una relación.
4. No puede haber 2 tuplas iguales en una relación.
5. Al igual que con las columnas, el orden de las filas es irrelevante, es decir, las tuplas no están ordenadas.

#### Tipos de relaciones más importantes:

1. **Relaciones base:** Son relaciones reales que tienen nombre y forman parte directa de la BD almacenada. Se corresponden con el nivel conceptual de la arquitectura ANSI.
2. **Vistas (o relaciones virtuales):** Son relaciones con nombre que se definen a partir de una consulta, no tienen datos almacenados, lo único que almacena es la definición de la consulta a realizar. Se corresponden con el nivel de visión o externo de la arquitectura ANSI.

#### Claves:

Una clave permite identificar a una fila de una tabla. En una tabla no hay filas repetidas, se identifican de un modo único mediante los valores de sus atributos. A veces la clave está formada por un único atributo, pero otras veces debe formarse por más de un atributo. Una clave debe cumplir 2 requisitos:

1. En cada fila de la tabla, el valor de la clave ha de identificarla de forma unívoca.
2. No se puede descartar ningún atributo de la clave para identificar la fila.

Una clave candidata de una relación (tabla) es el conjunto de atributos que identifican unívoca y mínimamente cada tupla de la relación. Es decir es una clave compuesta por el menor número de atributos y con atributos simples.

Una clave primaria o clave principal (primary key) es aquella clave candidata que el diseñador escoge para identificar las tuplas de la relación. No puede tener valores nulos.

Una clave alternativa: es cualquier clave candidata no escogida como clave primaria.

Una clave ajena (foreign key) de una tabla es el conjunto de atributos cuyos valores han de coincidir con los valores de la clave primaria de otra tabla. A través de las claves ajenas se construyen asociaciones entre tablas. Ambas claves deberán estar definidas sobre el mismo dominio y son muy importantes en el estudio de la integridad de datos del modelo relacional.

### ESQUEMA DE UNA BASE DE DATOS RELACIONAL:

Una base de datos relacional es un conjunto de relaciones normalizadas. Para representar su esquema deben darse:

1. El nombre de sus relaciones.
2. Los atributos de las relaciones.
3. Los dominios sobre los que se definen los atributos.
4. Las claves primarias y ajenas.

Las claves ajenas o foráneas se representan mediante diagramas referenciales.

---

Relación DEPARTAMENTOS		
CódigoDpto	NombreDpto	Presupuesto
D1	Marketing	1000,00
D2	Investigación	2500,50
D3	Desarrollo	5200,00

Relación EMPLEADOS			
IdEmpleado	NombreCompleto	Departamento	Salario
E1	Luis Pérez Sánchez	D2	1500,00
E2	Ana Vázquez Molino	D1	1500,00
E3	Paula Sanz Pino	D2	1650,00
E4	Felipe Ruiz Ferrer	D3	1450,00

El esquema completo de esta base de datos podría ser:

Relaciones, atributos, claves primarias (atributos subrayados):

DEPARTAMENTOS (CódigoDpto, NombreDpto, Presupuesto)

EMPLEADOS (IdEmpleado, NombreCompleto, Departamento, Salario)

Atributos y dominios:

CódigoDpto, IdEmpleado, Departamento: Conjunto de 2 caracteres.

NombreDpto, NombreCompleto: Conjunto de 40 caracteres.

Presupuesto, Salario: Número real de 4 cifras enteras y 2 cifras decimales.

Diagramas referenciales que muestran las asociaciones de relaciones y las claves ajenas:

Departamento  
EMPLEADOS —————> DEPARTAMENTOS: Departamento al que pertenece el empleado.

## RESTRICCIONES DEL MODELO RELACIONAL:

El modelo relacional impone 2 tipos de restricciones a tener en cuenta a la hora de diseñar una base de datos:

1. Restricciones inherentes al modelo relacional (algunas ya vistas como propiedades de las relaciones):
  - 1.1. En una relación no puede haber 2 tuplas iguales.
  - 1.2. El orden de las tuplas y de los atributos es irrelevante.
  - 1.3. Cada atributo de cada tupla sólo tiene un valor perteneciente al dominio al que corresponde.
  - 1.4. Ningún atributo que forme parte de la clave primaria puede tomar valor nulo.
2. Restricciones semánticas o de usuario:
  - 2.1. Restricción de la clave primaria (PRIMARY KEY en SQL): permite al usuario declarar uno o varios atributos como clave primaria en una relación.
  - 2.2. Restricción de unicidad (UNIQUE en SQL): permite al usuario definir claves alternativas, pues los atributos marcados como únicos no pueden repetirse.
  - 2.3. Restricción de obligatoriedad (NOT NULL en SQL): permite al usuario declarar si uno o varios atributos no pueden tomar valores nulos, por tanto, deben tener siempre un valor.

2.4. Restricción de clave ajena (FOREIGN KEY en SQL) o integridad referencial: El usuario la utiliza para asociar relaciones de una BD mediante claves ajenas. La integridad referencial indica que los valores de la clave ajena de esa relación se corresponden obligatoriamente con los valores de la clave primaria de otra relación. Tras definir las claves ajenas hay que tener en cuenta qué ocurrirá cuando se borren (ON DELETE en SQL) o se modifiquen (ON UPDATE en SQL) tuplas de la relación cuya clave primaria se corresponde con esta clave ajena.

2.4.1. Borrado y/o modificación en cascada (CASCADE en SQL): Si el usuario especifica esta restricción, el borrado o modificación de una tupla en una relación ocasiona un borrado o modificación de las tuplas de otras relaciones asociadas a ésta, cuyas claves ajenas se correspondan con su clave primaria.

- *Por ejemplo, si se borra un departamento de la relación DEPARTAMENTOS, se borrarán todos los empleados de la tabla EMPLEADOS que pertenezcan a ese departamento borrado; y si se modifica el código de un departamento de la relación DEPARTAMENTOS, automáticamente cambiarán todos los valores del campo Departamento en la relación EMPLEADOS que tenía el mismo código al nuevo valor ya modificado.*

2.4.2. Borrado y/o modificación restringidos (RESTRICT en SQL): Si el usuario especifica esta restricción, el borrado o modificación de una tupla en una relación será imposible si existen tuplas de otras relaciones asociadas a ésta, cuyas claves ajenas se correspondan con su clave primaria.

- *Así, no se podría, por ejemplo, eliminar un departamento si tiene empleados asociados a él, ni tampoco se podría cambiar su código si ya tenía empleados; por tanto, únicamente podrán realizarse esas operaciones en departamentos sin empleados asociados.*

2.4.3. Borrado y/o modificación con puesta a nulos (SET NULL en SQL): Si el usuario especifica esta restricción, el borrado o modificación de una tupla en una relación ocasiona la puesta a NULL de la clave ajena de las tuplas de otras relaciones asociadas a ésta, cuyas claves ajenas se correspondían con su clave primaria.

2.4.4. Borrado y/o modificación con puesta a valor por defecto (SET DEFAULT en SQL): Si el usuario especifica esta restricción, el borrado o modificación de una tupla en una relación ocasiona la puesta al valor por defecto especificado de la clave ajena de las tuplas de otras relaciones asociadas a ésta, cuyas claves ajenas se correspondían con su clave primaria.

2.5. Restricción de verificación (CHECK en SQL): Esta restricción permite al usuario especificar condiciones que deban cumplir los valores de los atributos. Cada vez que se realiza una inserción o una actualización de datos se comprueba si los valores cumplen la condición, rechazando la operación si no la cumplen.

- *Un ejemplo en el SQL de Oracle de creación de tablas relacionadas con restricciones sería:*

```
CREATE TABLE FABRICANTES(  
    CD_FAB NUMBER(3) NOT NULL DEFAULT 100,  
    NOMBRE VARCHAR2(15) UNIQUE,  
    PAIS VARCHAR2(15) CONSTRAINT CK_PA CHECK(PAIS=UPPER(PAIS)),  
    CONSTRAINT PK_FA PRIMARY KEY (CD_FAB),  
    CONSTRAINT CK_NO CHECK(NOMBRE=UPPER(NOMBRE))  
);  
CREATE TABLE ARTICULOS(  
    ARTIC VARCHAR2(20) NOT NULL,  
    COD_FA NUMBER(2) NOT NULL,  
    PESO NUMBER(3) NOT NULL CONSTRAINT CK1_AR CHECK(PESO>0),
```



```

CATEGORIA VARCHAR2(10) NOT NULL,
PRECIO_VENTA NUMBER(4) CONSTRAINT CK2_AR CHECK(PRECIO_VENTA>0),
PRECIO_COSTO NUMBER(4) CONSTRAINT CK3_AR CHECK(PRECIO_COSTO>0),
EXISTENCIAS NUMBER(5),
CONSTRAINT PK_ART PRIMARY KEY (ARTIC,COD_FA,PESO,CATEGORIA),
CONSTRAINT FK_ARFA FOREIGN KEY (COD_FA) REFERENCES FABRICANTES
(CD_FAB) ON DELETE CASCADE,
CONSTRAINT CK_CAT CHECK(CATEGORIA IN ('Primera','Segunda','Tercera'))
);

```

2.6. Aserciones (ASSERTION en SQL): Igual que CHECK pero puede afectar a 2 o más relaciones, por tanto, la condición a cumplir se establece sobre campos de distintas relaciones. Pueden implicar a subconsultas en la condición.

2.7. Disparadores (TRIGGER): Las restricciones anteriores son declarativas, sin embargo este tipo es procedimental. El usuario podrá programar una serie de acciones distintas ante una determinada condición.

- *Ejemplo: Un disparador que puede realizar el usuario para auditar las operaciones de modificación y borrado de datos de la tabla EMPLE, de forma que cada vez que se realiza una operación de actualización o borrado se inserta en la tabla AUDITAEMPLE una fila que contendrá varios campos: fecha y hora de la operación, número y apellido del empleado afectado, y la operación que se realiza.*

```

CREATE OR REPLACE TRIGGER auditar_act_emp
BEFORE INSERT OR DELETE ON EMPLE FOR EACH ROW
BEGIN
  IF DELETING THEN
    INSERT INTO AUDITAREMPLE VALUES
      (TO_CHAR(sysdate,'DD/MM/YY*HH24:MI*') || :OLD.EMP_NO || '*' || :OLD.APELLIDO || '*BORRADO');
  ELSIF INSERTING THEN
    INSERT INTO AUDITAREMPLE VALUES
      (TO_CHAR(sysdate,'DD/MM/YY*HH24:MI*') || :NEW.EMP_NO || '*' || :NEW.APELLIDO || '*INSERCIÓN');
  END IF;
END;

```

## Ejemplo práctico con SQL:

Una empresa desea almacenar información sobre sus departamentos y sus empleados para lo cual dispone de sendas tablas DEPT y EMP. Los atributos de la tabla DEPT son: deptno (número de departamento que actúa como clave primaria), dnombre (nombre de departamento) y loc (ciudad donde está ubicado). Los atributos de la tabla EMP son: empno (número de empleado que actúa como clave primaria), enombre (nombre), puesto (actividad o responsabilidad que desempeña dentro de la empresa: presidente, vendedor, etc), fecha (fecha de alta en la empresa), sal (salario o sueldo), comm (comisiones por venta) y deptno (departamento al que pertenece el empleado; clave ajena).

Han de tenerse en cuenta las siguientes restricciones:

1. Los puestos laborales existentes en la empresa son ADMINISTRATIVO, VENDEDOR, GERENTE, ANALISTA y PRESIDENTE.
2. Los vendedores son los únicos empleados que pueden tener comisiones.
3. Las comisiones de un vendedor no pueden superar el 200 % de su salario.

### Creación de la tabla de departamentos

```

CREATE TABLE dept(
deptno NUMBER(2) PRIMARY KEY,
dnombre VARCHAR(15),
loc CHAR(15));

```

**Creación de la tabla de empleados**

```
CREATE TABLE emp(  
empno NUMBER(4) PRIMARY KEY,  
enombre VARCHAR(20),  
puesto VARCHAR(15),  
fecha DATE,  
sal NUMBER(7),  
comm NUMBER(7),  
deptno NUMBER(2),  
FOREIGN KEY (deptno) REFERENCES dept,  
CONSTRAINT ck_puesto CHECK (puesto IN  
( 'ADMINISTRATIVO', 'VENDEDOR', 'GERENTE', 'ANALISTA', 'PRESIDENTE' )),  
CONSTRAINT ck_comm1 CHECK ((comm IS NULL AND puesto != 'VENDEDOR') OR  
(comm IS NOT NULL AND puesto = 'VENDEDOR' AND comm>=0)),  
CONSTRAINT ck_comm2 CHECK (comm<=2*sal));
```

También se puede crear la tabla sin las restricciones CONSTRAINT y especificar éstas posteriormente mediante ALTER TABLE. Por ejemplo:

```
ALTER TABLE emp ADD CONSTRAINT ck_comm2 CHECK (comm<=2*sal);
```

**Creación de restricciones genéricas adicionales**

*Un departamento no puede tener más de un gerente.*

```
CREATE ASSERTION as_gerente CHECK (NOT EXISTS (  
SELECT COUNT(*) FROM emp WHERE puesto='GERENTE'  
GROUP BY deptno HAVING COUNT(*) > 1));
```

*Solo puede existir un presidente en la empresa.*

```
CREATE ASSERTION as_presidente CHECK (  
(SELECT COUNT(*) FROM emp WHERE puesto='PRESIDENTE') < 2);
```

**Insertión de tuplas de ejemplo**

*Insertión de filas en la tabla de departamentos.*

```
INSERT INTO dept VALUES(10, 'CONTABILIDAD', 'SEVILLA');  
INSERT INTO dept VALUES(20, 'I+D', 'MADRID');  
INSERT INTO dept VALUES(40, 'OPERACIONES', 'BARCELONA');  
INSERT INTO dept VALUES(30, 'VENTAS', 'MALAGA');
```

*Insertión de filas en la tabla de empleados.*

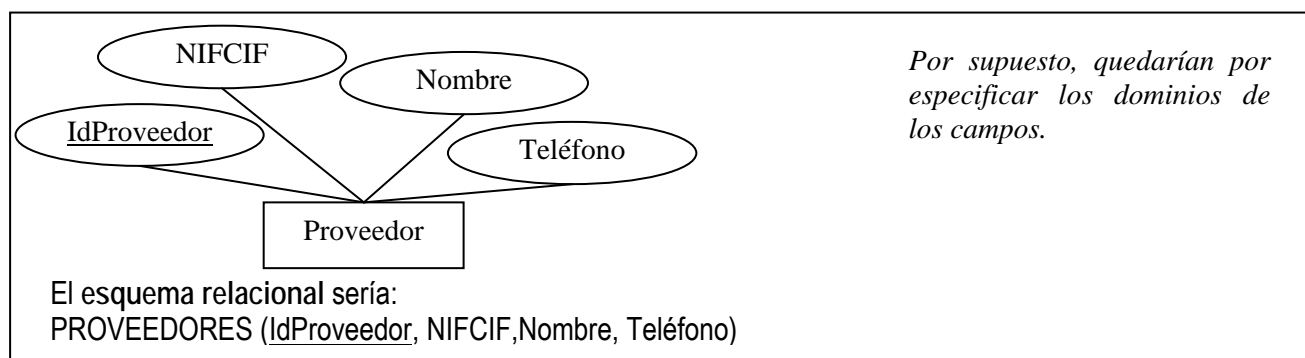
```
INSERT INTO emp VALUES(7369, 'PEREZ PEREZ', 'ADMINISTRATIVO', 171280, 150000, NULL, 20);  
INSERT INTO emp VALUES(7499, 'LOPEZ CASTO', 'VENDEDOR', 200281, 150000, 300000, 30);  
INSERT INTO emp VALUES(7521, 'RUIZ REY', 'VENDEDOR', 220281, 125000, 275000, 30);  
INSERT INTO emp VALUES(7566, 'BARRIO PADILLA', 'GERENTE', 020481, 300000, NULL, 20);  
INSERT INTO emp VALUES(7654, 'JIMENEZ MARTIN', 'VENDEDOR', 280981, 250000, 150000, 30);  
INSERT INTO emp VALUES(7698, 'SALVADOR BRAVO', 'GERENTE', 010581, 350000, NULL, 30);  
INSERT INTO emp VALUES(7782, 'HALCON MALTES', 'GERENTE', 090681, 350000, NULL, 10);  
INSERT INTO emp VALUES(7788, 'PERA LOPEZ', 'ANALISTA', 091181, 245000, NULL, 20);  
INSERT INTO emp VALUES(7839, 'GONZALEZ MARQUEZ', 'PRESIDENTE', 171181, 500000, NULL, 10);  
INSERT INTO emp VALUES(7844, 'ROBLES ROBLES', 'VENDEDOR', 080981, 150000, 280000, 30);  
INSERT INTO emp VALUES(7876, 'NADAL SILES', 'ADMINISTRATIVO', 230981, 160000, NULL, 20);  
INSERT INTO emp VALUES(7900, 'NARANJO DIAZ', 'ADMINISTRATIVO', 031281, 140000, NULL, 30);  
INSERT INTO emp VALUES(7902, 'ROMERO BORT', 'ANALISTA', 031281, 245000, NULL, 20);  
INSERT INTO emp VALUES(7934, 'MILLAN VAZQUEZ', 'ADMINISTRATIVO', 230182, 140000, NULL, 10);
```

## TRANSFORMACIÓN DE UN DIAGRAMA ENTIDAD-RELACIÓN A UN ESQUEMA RELACIONAL:

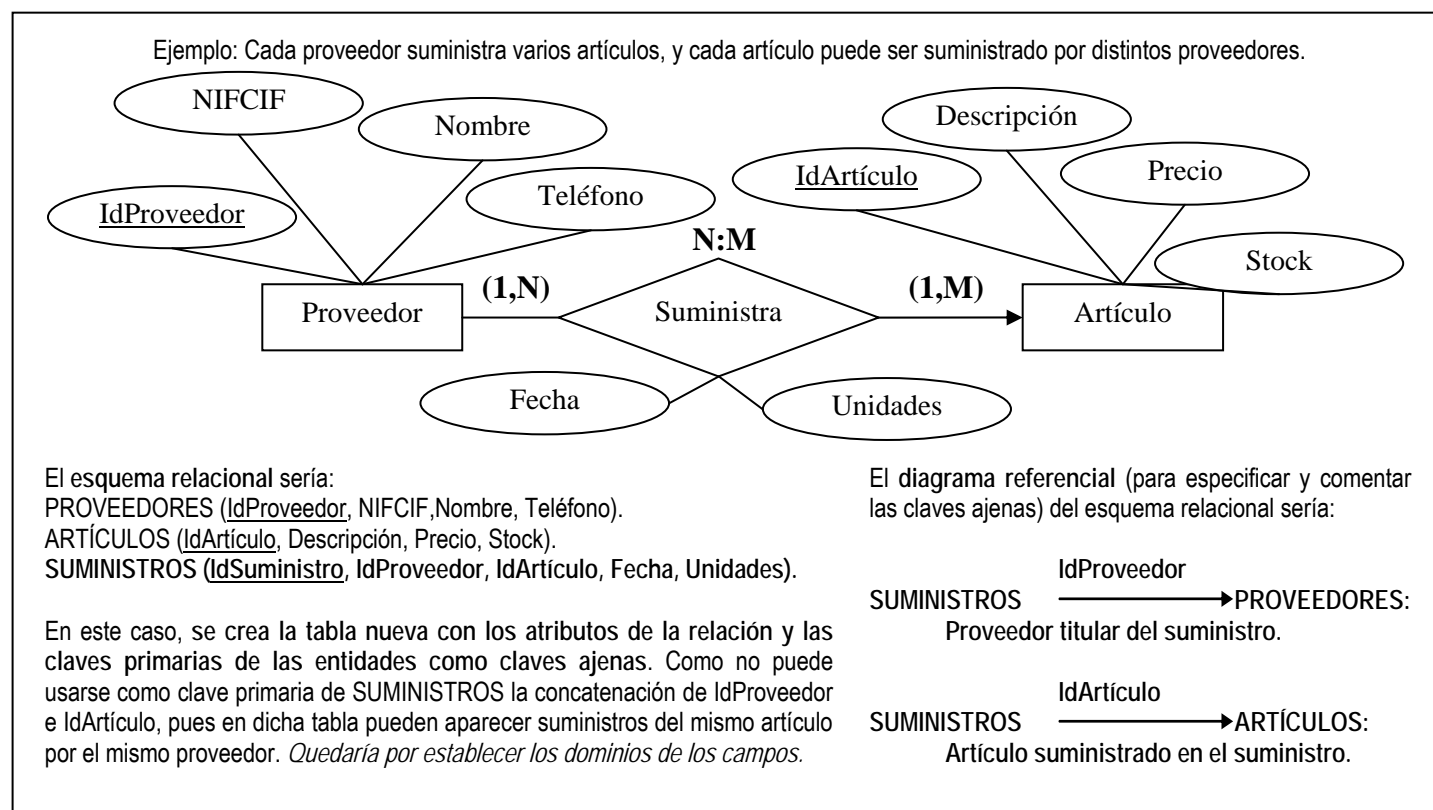
Una vez obtenido el esquema conceptual mediante un diagrama E-R, puede definirse el modelo lógico de datos mediante un esquema relacional.

### 1 - Las reglas básicas para transformar un diagrama E-R a un esquema relacional son:

1. Toda entidad se transforma en una tabla.
2. Todo atributo de la entidad se transforma en columna de la tabla.
3. La clave primaria de la entidad se transforma en la clave primaria de la tabla.



4. Toda relación N:M se transforma en una tabla, que tendrá como claves ajenas las claves primarias de las entidades que asocia. Y podrá tener como clave primaria la concatenación de los atributos clave de las entidades que asocia si es posible, si no, se utilizan junto con uno o varios atributos de la relación o se le agrega un campo identificador nuevo como clave primaria.



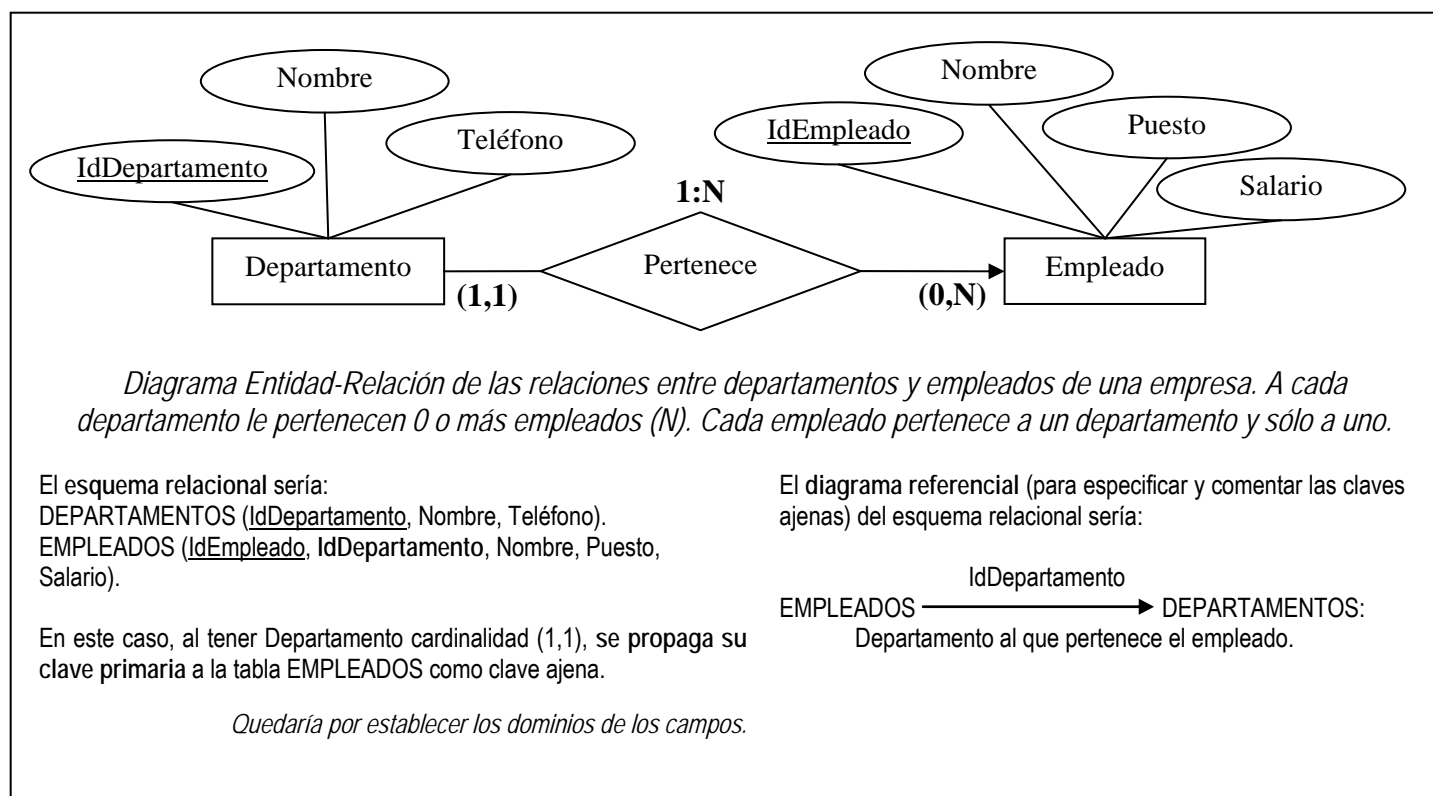
5. Para las relaciones 1:N existen 2 soluciones:

5.1. **Propagar la clave**, es decir, se propagan la clave primaria de la entidad con cardinalidad máxima 1 y los atributos de la propia relación a la entidad cuya cardinalidad máxima es N. Se suele realizar cuando:

- La cardinalidad de la entidad cuya cardinalidad máxima es 1 es (1,1), es decir, es obligatoria.

5.2. **Transformar la relación en una tabla** (exactamente igual a como ocurre con las relaciones N:M). Se puede realizar cuando ocurren algunos de estos casos:

- La cardinalidad de la entidad cuya cardinalidad máxima es 1 es (0,1), es decir, es opcional.
- Se prevé que se convertirá en una relación N:M.
- La relación tiene atributos propios.



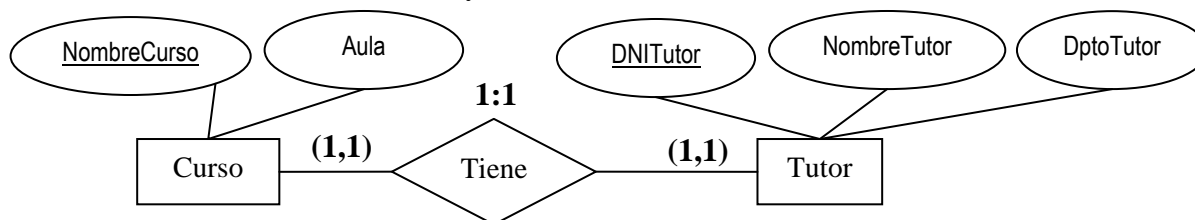
6. Para las relaciones 1:1 se tienen en cuenta las cardinalidades de las entidades que participan en la relación.

6.1. **Unir ambas entidades en una tabla**, cuando ambas entidades tienen cardinalidad (1,1). Se escoge como clave primaria de la tabla a una cualquiera de las dos claves primarias de las entidades.

6.2. **Propagar la clave**, cuando una entidad tiene cardinalidad (1,1) y la otra (0,1). Se propaga la clave primaria de la entidad con cardinalidad (1,1) a la tabla resultante de la entidad con cardinalidad (0,1) convirtiéndose en clave ajena.

6.3. **Transformar la relación en una tabla**, cuando ambas entidades tienen cardinalidad (0,1). Se transforma la relación en una tabla independiente, tal y como se hacía cuando la relación era N:M.

Cada curso tiene un único tutor, y cada tutor lo es únicamente de un curso.



El esquema relacional sería:

CURSOSYTUTORES (NombreCurso, Aula, DNITutor, NombreTutor, DptoTutor).

En este caso, al tener ambas entidades cardinalidad (1,1), se funden las 2 entidades en una sola tabla, y se toma como clave primaria a cualquiera de las claves de las entidades, en este caso, se eligió la clave NombreCurso.

*Quedaría por establecer los dominios de los campos.*

## 2 - Transformación de otros elementos del modelo E-R:

1. Relaciones reflexivas: Son las relaciones binarias en las que únicamente participa un tipo de entidad. Pueden encontrarse los siguientes casos:

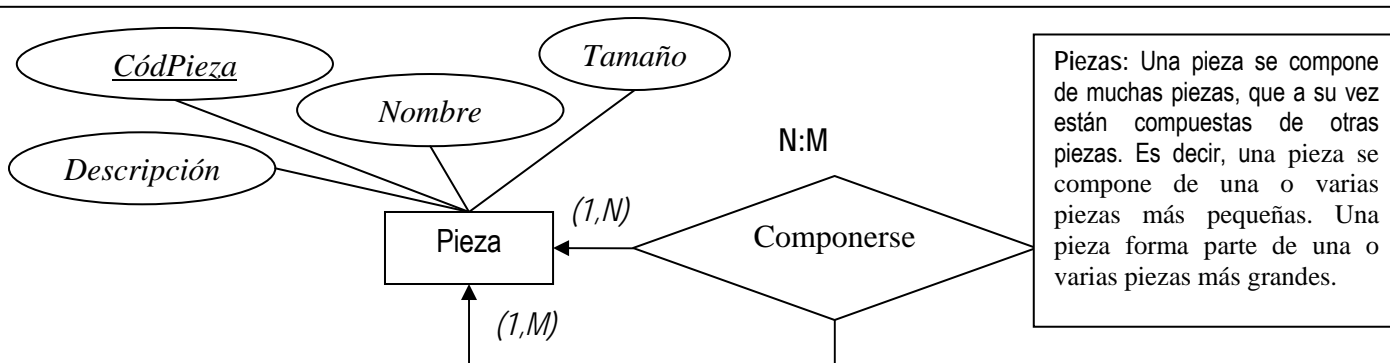
- 1.1. Si la relación es 1:1, no se crea una segunda tabla, si no que en la tabla resultante se agregará 2 veces el mismo atributo, como clave primaria y como clave ajena a ella misma.

- 1.2. Si la relación es 1:N, hay que ver 2 casos:

- 1.2.1. Si la entidad tiene cardinalidad (1,N) se procede como en el caso de relaciones 1:1.

- 1.2.2. Si la entidad no es obligatoria, es decir, tiene cardinalidad (0,N), se crea una nueva tabla cuya clave será la de la entidad del lado muchos, y además se propaga la clave a la nueva tabla como clave ajena.

- 1.3. Si la relación es N:M, se trata igual que en las relaciones binarias. La tabla resultante de la relación contendrá 2 veces la clave primaria de la entidad del lado muchos, más los atributos de la relación si los hubiera. La clave de esta nueva tabla será la combinación de las 2.



El esquema relacional sería:

PIEZAS (CódPieza, Descripción, Tamaño, Nombre).

COMPONENTES (CódPieza, CódPiezaComponente).

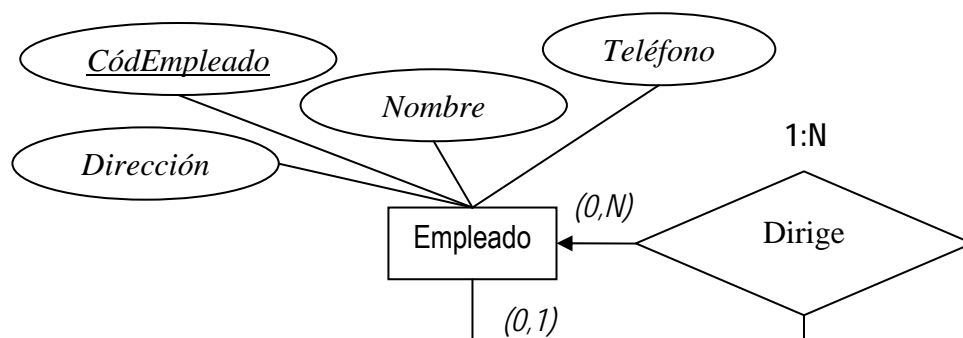
En este caso, se crea la tabla COMPONENTES que tiene como clave primaria la concatenación de la clave primaria de la tabla PIEZAS (identificando la pieza compuesta), más el campo CódPiezaComponente también (identificando la pieza de la que se compone la pieza compuesta). Además son ambas claves ajenas.

El diagrama referencial (para especificar y comentar las claves ajenas) del esquema relacional sería:

COMPONENTES  $\xrightarrow{\text{CódPieza}}$  PIEZAS: Código de la pieza deseada que se quiere fabricar a partir de otras.

COMPONENTES  $\xrightarrow{\text{CódPiezaComponente}}$  PIEZAS: Código de la pieza menor que hace falta para fabricar la pieza deseada.

**Empleados y empleados jefes:** Un empleado puede dirigir a muchos empleados si es el jefe, o a ninguno si no es el jefe. Un empleado es dirigido por un jefe, o por ninguno si él mismo es el jefe.



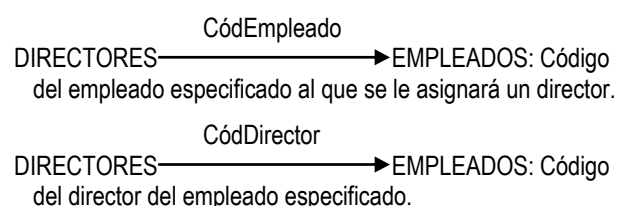
El esquema relacional sería:

EMPLEADOS (CódEmpleado, Dirección, Teléfono, Nombre).

DIRECTORES (CódEmpleado, CódDirector)

En este caso, se crea la tabla DIRECTORES que tiene a la vez como clave primaria y ajena la misma que la de la tabla EMPLEADOS, identificando al empleado al que se hace referencia; y además, tiene el mismo campo como clave ajena que indica qué otro empleado es su director.

El diagrama referencial (para especificar y comentar las claves ajenas) del esquema relacional sería:



## 2. Generalizaciones o especializaciones: Las diferentes opciones para la transformación de jerarquías del modelo E-R al modelo relacional son:

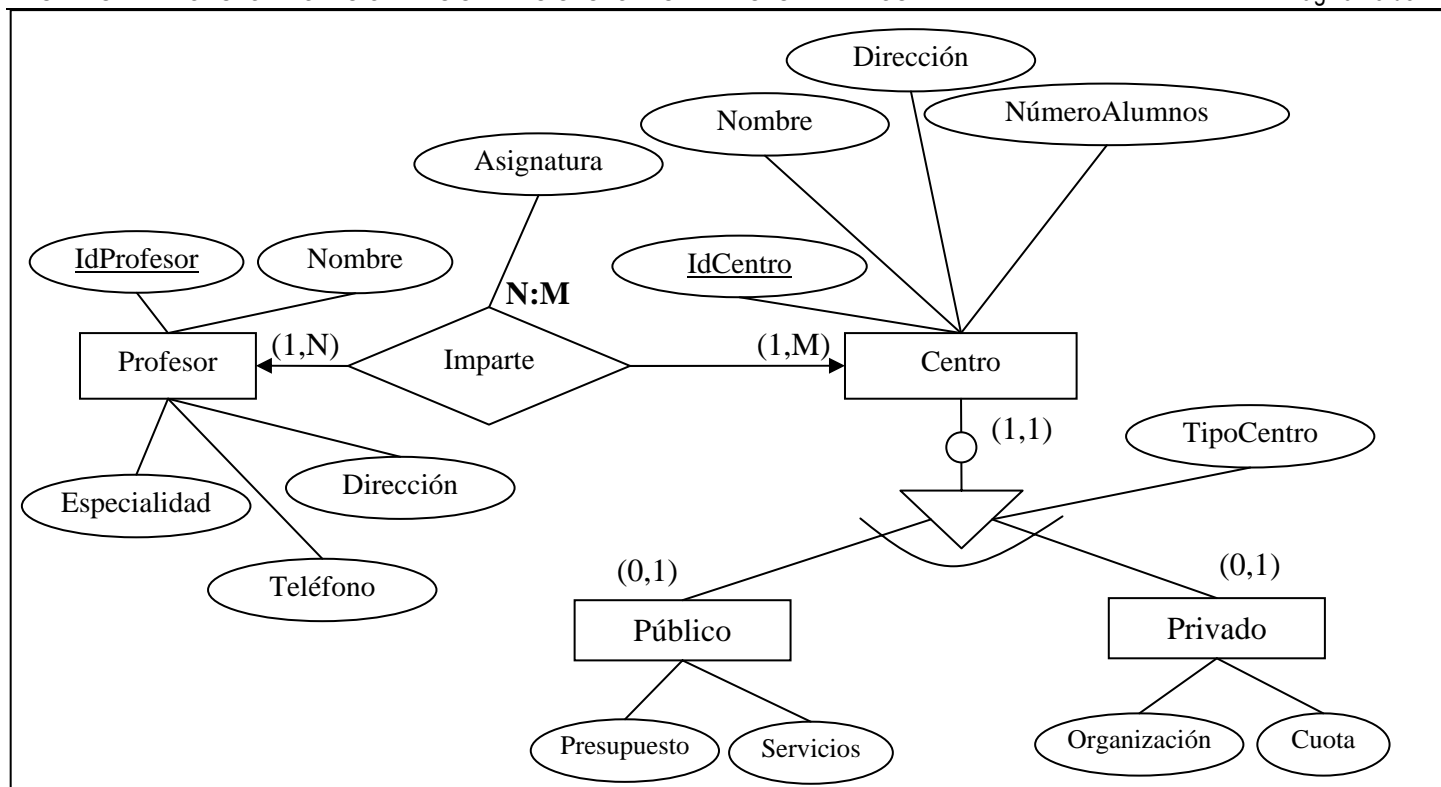
2.1. Integrar todas las entidades en una única tabla absorbiendo los subtipos: Se crea una tabla que contiene todos los atributos del supertipo, todos los de los subtipos, y el atributo discriminatorio para distinguir a qué subtipo pertenece cada registro de la tabla. Esta regla puede aplicarse a cualquier tipo de jerarquía, es muy simple de realizar, pero genera demasiados valores nulos en los atributos opcionales propios de cada subtipo.

2.2. Eliminación del supertipo en jerarquías totales y exclusivas: Transfiriendo los atributos del supertipo a cada uno de los subtipos, creándose una tabla por cada subtipo, el supertipo no tendrá tabla, y se elimina el atributo que distingue entre subtipos. Se crea redundancia en la información pues los atributos del supertipo se repiten en cada uno de los subtipos. El número de relaciones aumenta, pues las relaciones del supertipo pasan a cada uno de los subtipos.

2.3. Insertar una relación 1:1 entre el supertipo y los subtipos: Los atributos se mantienen y cada subtipo se identificará con una clave ajena referenciando a la clave primaria del supertipo. El supertipo mantendrá una relación 1:1 con cada subtipo.

### Ejemplo de los Profesores y los centros especializados en públicos y privados:

- Hay profesores que *imparten* clases en 2 tipos de centros educativos: públicos y privados.
- Un profesor puede impartir clase en varios centros, ya sean públicos o privados.
- La asignatura será un atributo de la relación entre el profesor y el centro donde imparte.
- Los centros educativos sólo pueden ser públicos o privados.
- Un centro público no puede ser privado a la vez, ni a la inversa.
- Los atributos específicos para los centros públicos son: el presupuesto y los servicios; y para los privados son: la organización y la cuota.



**SOLUCIÓN:** (Aparecen en negrita las principales diferencias entre los distintos métodos, subrayados los campos pertenecientes a la clave primaria, y en cursiva los campos que son clave ajena; no se especificarán los diagramas referenciales ni los dominios de los campos).

1. Integrar todas las entidades en una única tabla absorbiendo los subtipos:

PROFESORES(IdProfesor, Nombre, Dirección, Teléfono, Especialidad).  
 CENTROS(IdCentro, Nombre, Dirección, NúmeroAlumnos, TipoCentro, Servicios, Presupuesto, Organización, Cuota).  
 IMPARTICIONES(IdProfesor, IdCentro, Asignatura).

2. Eliminación del supertipo en jerarquías totales y exclusivas:

PROFESORES(IdProfesor, Nombre, Dirección, Teléfono, Especialidad).  
 PÚBLICOS(IdCentro, Nombre, Dirección, NúmeroAlumnos, Servicios, Presupuesto).  
 PRIVADOS(IdCentro, Nombre, Dirección, NúmeroAlumnos, Organización, Cuota).  
 IMPARTICIONES\_PÚBLICOS(IdProfesor, IdCentro, Asignatura).  
 IMPARTICIONES\_PRIVADOS(IdProfesor, IdCentro, Asignatura).

3. Insertar una relación 1:1 entre el supertipo y los subtipos:

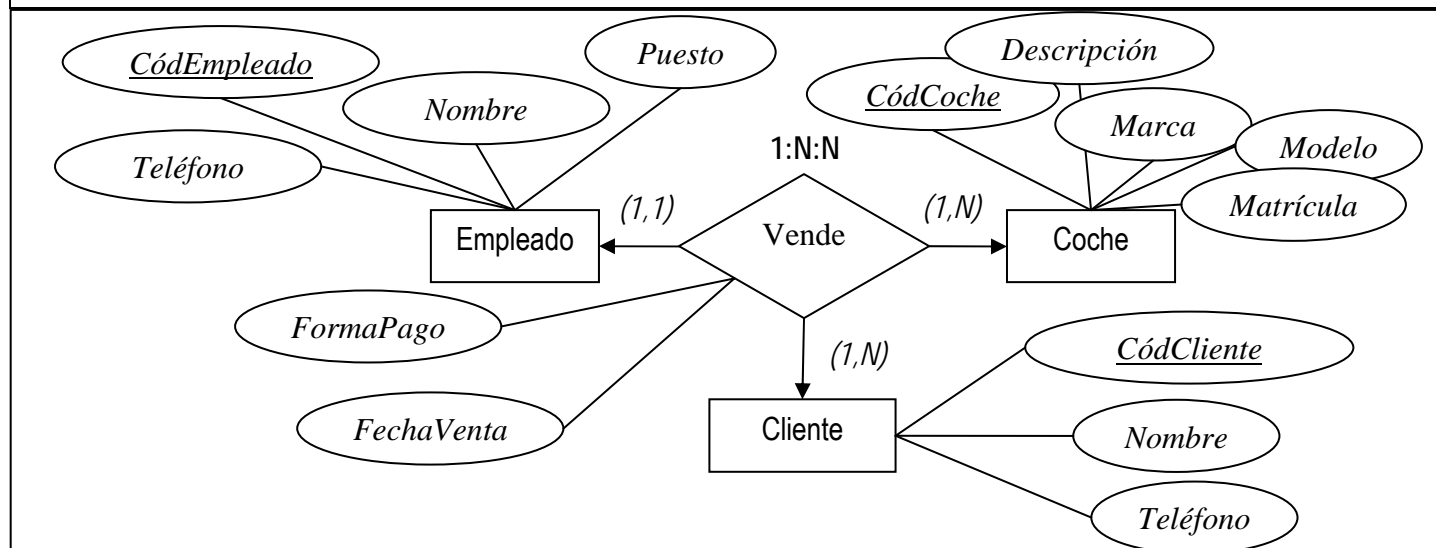
PROFESORES(IdProfesor, Nombre, Dirección, Teléfono, Especialidad).  
 CENTROS(IdCentro, Nombre, Dirección, NúmeroAlumnos, TipoCentro).  
 PÚBLICOS(IdCentro, Servicios, Presupuesto).  
 PRIVADOS(IdCentro, Organización, Cuota).  
 IMPARTICIONES(IdProfesor, IdCentro, Asignatura).

3. Relaciones N-arias (ternarias, cuaternarias, etc.): En este tipo de relaciones se asocian 3 ó más entidades. Se pasan todas las entidades a tablas tal cual. La relación también se convierte a una tabla, que va a contener sus atributos más las claves primarias de todas las entidades que asocia como claves ajenas. Hay 2 casos:
- 3.1. Si la relación es N:N:N, es decir, todas las entidades participan con cardinalidad máxima N, la clave de la tabla resultante de la relación es la unión de las claves ajenas que referencian a las entidades que asocia.
  - 3.2. Si la relación es 1:N:N, es decir, una sola entidad participa con cardinalidad máxima 1, la clave de la tabla resultante de la relación es la unión de las claves ajenas que referencian a las entidades que asocia.

excepto la de la entidad que participa con cardinalidad máxima 1, que queda como un atributo más y como clave ajena, pero no formará parte de la clave primaria de dicha tabla resultante de la relación.

#### Vendedores de coches (empleados, clientes y coches):

- En una tienda de coches, un empleado vende coches a sus clientes.
- En cada venta, un único empleado puede vender varios coches a varios clientes.
- En una operación de venta hay que tener en cuenta la forma de pago y la fecha de venta.



El resultado en el modelo relacional (siendo una relación ternaria del tipo 1:N:N) es:

EMPLEADOS(CódEmpleado,Nombre,Puesto,Teléfono).

COCHES(CódCoche,Descripción,Marca,Modelo,Matrícula).

CLIENTES(CódCliente,Nombre,Teléfono).

VENTAS(CódCoche,CódCliente,CódEmpleado,FormaPago,FechaVenta).

### NORMALIZACIÓN DE ESQUEMAS RELACIONALES:

El proceso de normalización de una base de datos consiste en aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo E-R (entidad-relación) al modelo relacional que elimine las dependencias no deseadas entre los atributos.

Las bases de datos relacionales se normalizan para:

1. Evitar la redundancia de los datos.
2. Evitar problemas de actualización (tras insertar, modificar o borrar) de los datos en las tablas.
3. Proteger la integridad de los datos.

La normalización se lleva a cabo en una serie de pasos, llamados formas normales, que van reconstruyendo las tablas haciéndolas más robustas y menos vulnerables a las anomalías que pudiesen surgir tras una actualización.

**Dependencia funcional:** Una dependencia funcional es una relación o conexión entre uno o más atributos de la misma tabla. Por ejemplo si conocemos el valor de FechaDeNacimiento podemos conocer el valor de Edad. Las dependencias funcionales se escriben utilizando una flecha, de la siguiente manera:

FechaDeNacimiento → Edad



Aquí al atributo FechaDeNacimiento se le conoce como un determinante, pues para cada valor de FechaDeNacimiento hay un valor único valor de Edad asociado. Se puede leer de dos formas: FechaDeNacimiento determina a Edad o Edad es funcionalmente dependiente de FechaDeNacimiento. Se dice que Edad es completamente dependiente de FechaDeNacimiento si depende funcionalmente de ese atributo y no depende de ningún otro de la tabla.

**Dependencia funcional transitiva:** Supongamos, por ejemplo, que en una relación los estudiantes solo pueden estar matriculados en un solo curso y supongamos que los profesores solo pueden dar un curso.

ID\_Estudiante  $\rightarrow$  Curso

Curso  $\rightarrow$  Profesor\_Asignado

ID\_Estudiante  $\rightarrow$  Curso  $\rightarrow$  Profesor\_Asignado (Dependencia funcional transitiva)

Entonces tenemos que ID\_Estudiante determina a Curso y el Curso determina a Profesor\_Asignado; indirectamente podemos saber a través del ID\_estudiante el Profesor\_Asignado. Entonces tenemos una dependencia transitiva.

De la normalización (lógica) a la implementación (física o real) puede ser sugerible tener en cuenta éstas dependencias funcionales para lograr mayor eficiencia en las tablas construidas.

**Formas Normales:** Las tres primeras formas normales son suficientes para cubrir las necesidades de la mayoría de las bases de datos, las demás son opcionales. El creador de estas 3 primeras formas normales (o reglas) fue Edgar F. Codd, éste introdujo la normalización en un artículo llamado A Relational Model of Data for Large Shared Data Banks Communications of the ACM, Vol. 13, No. 6, June 1970, pp. 377-387.

**Primera Forma Normal (1FN):** Se eliminan los grupos repetitivos. Los atributos han de ser atómicos, es decir, cada atributo de la tabla toma un único valor del dominio correspondiente. Gráficamente las celdas de la tabla contienen solo un valor, en cada uno de los atributos sólo se puede incluir un dato, aunque sea compuesto, pero no se puede incluir una lista de datos. Se trata de que cada atributo guarde la menor cantidad de información posible.

Para eliminar los grupos repetitivos puede ponerse cada a uno de ellos en una tabla aparte, esa nueva tabla hereda la clave primaria de la relación en la que se encontraban.

**Segunda Forma Normal (2FN):** Dependencia completa. Una tabla está en 2FN si y sólo si está en 1FN y si sus atributos no principales (que no pertenecen a la clave primaria) dependen de forma completa de la clave primaria (de todos los atributos de la clave primaria). Se aplica en tablas con claves primarias compuestas por varios atributos, por tanto, toda tabla que tenga como clave primaria sólo un atributo está en 2FN si ya lo estaba en 1FN.

Teniendo la tabla T(A, B, C, D), si C depende funcionalmente sólo de A ( $A \rightarrow C$ ) y D depende funcionalmente de A y B ( $A, B \rightarrow D$ ) entonces debe dividirse la tabla en 2 tablas: T1 (A, C) y T2 (A, B, D). Así se eliminaron las dependencias parciales de la clave primaria en la tabla T. T1 y T2 ya están en 2FN, pues los atributos que no forman parte de la clave primaria dependen de todos los atributos que la componen.

**Tercera Forma Normal (3FN):** Se eliminan las dependencias transitivas. Una tabla está en 3FN si y sólo si está en 2FN y todo atributo que no está en la clave primaria no depende transitivamente de la clave primaria.

---

El valor de esta columna debe depender directamente de la clave. Todos los valores de una tabla deben identificarse únicamente por la clave directamente, y no por un campo intermedio no principal de la tabla que a su vez depende funcionalmente de la clave principal (dependencia transitiva).

Teniendo la tabla T(A,B,C,D), si  $A \rightarrow B$ ,  $A \rightarrow C$ , y  $C \rightarrow D$ , vemos que D no depende de forma directa de la clave primaria simple A, pero sí depende transitivamente de ella a través del campo C. La tabla debe descomponerse en 2 tablas: T1(A,B,C) y T2(C,D). Así se eliminaron las dependencias transitivas en las tablas.

Las formas normales opcionales son: Forma normal de Boyce-Codd (FNBC), Cuarta Forma Normal (4FN) y Quinta Forma Normal (5FN).

Ejemplo: Pasar a tercera forma normal la siguiente tabla basada en el modelo relacional:

RELACIÓN EMPLEADOS

<u>IdEmpleado</u>	<u>Teléfono</u>	<u>Código Postal</u>	<u>Localidad</u>	<u>Cargo</u>	<u>Dpto</u>
1	957894511 642889978	41013	Sevilla	Administrativo	Administración
2	910022354 910022355 609415004	28005	Madrid	Informática	Técnico
3	956448877	11005	Cádiz	Auxiliar Administrativo	Administración

Para que esté en 1FN hay que eliminar los grupos repetitivos, es decir, que en cada celda de la tabla hay un único valor y no varios, como se observa en el campo Teléfono.

Así que sacaremos el campo con valores múltiples en cada celda de la tabla, separando los valores en distintas filas en otra tabla, que tendrá como clave primaria compuesta la clave primaria de la tabla original más el atributo con valores múltiples.

RELACIÓN EMPLEADOS

<u>IdEmpleado</u>	<u>Código Postal</u>	<u>Localidad</u>	<u>Cargo</u>	<u>Dpto</u>
1	41013	Sevilla	Administrativo	Administración
2	28005	Madrid	Informático	Técnico
1	11005	Cádiz	Auxiliar Administrativo	Administración

RELACIÓN TELÉFONOS

<u>IdEmpleado</u>	<u>Teléfono</u>	<u>Cargo</u>
1	957894511	Administrativo
1	642889978	Administrativo
2	910022354	Informático
2	910022355	Informático
2	609415004	Informático
1	956448877	Auxiliar Administrativo

Ya está en 1FN. Para pasar el resultado a 2FN, como ya está en 1FN, todos los campos de la tabla que no formen parte de la clave primaria deben depender de todos los atributos de la clave primaria a la vez, y no sólo de unos pocos de la clave primaria pero no de otros.

Los campos Localidad y Código Postal dependen a la vez de Nombre y de Cargo, en cambio el campo Dpto si depende exclusivamente de Cargo y no de Nombre. Por tanto se han detectado dependencias funcionales parciales, pues algunos atributos dependen de algunos campos de la clave primaria pero no de todos. La relación TELÉFONOS ya está en 2FN al no tener atributos que no pertenezcan a la clave primaria, pero la relación EMPLEADOS deberá dividirse para evitar las dependencias funcionales parciales:

RELACIÓN CARGOS

Cargo	Dpto
Administrativo	Administración
Informático	Técnico
Auxiliar Administrativo	Administración

RELACIÓN EMPLEADOS

IdEmpleado	Código Postal	Localidad	Cargo
1	41013	Sevilla	Administrativo
2	28005	Madrid	Informático
1	11005	Cádiz	Auxiliar Administrativo

RELACIÓN TELÉFONOS

IdEmpleado	Teléfono	Cargo
1	957894511	Administrativo
1	642889978	Administrativo
2	910022354	Informática
2	910022355	Informática
2	609415004	Informática
1	956448877	Auxiliar Administrativo

Ya está en 2FN. Para pasar el resultado a 3FN, como ya está en 2FN, todos los campos de la tabla que no formen parte de la clave primaria deben depender "directamente" de todos los atributos de la clave primaria al completo, y no de otros atributos que a su vez dependan de la clave primaria.

Vemos que en la relación EMPLEADOS, el campo Localidad depende realmente del Código Postal, que a su vez depende de la clave primaria, por tanto, se da una dependencia funcional transitiva, y esto obligará a dividir la tabla:

RELACIÓN CARGOS

Cargo	Dpto
Administrativo	Administración
Informático	Técnico
Auxiliar Administrativo	Administración

RELACIÓN EMPLEADOS

IdEmpleado	Código Postal	Cargo
1	41013	Administrativo
2	28005	Informático
1	11005	Auxiliar Administrativo

RELACIÓN CÓDIGOSPOSTALES

Código Postal	Localidad
41013	Sevilla
28005	Madrid
11005	Cádiz

RELACIÓN TELÉFONOS

IdEmpleado	Teléfono	Cargo
1	957894511	Administrativo
1	642889978	Administrativo
2	910022354	Informática
2	910022355	Informática
2	609415004	Informática
1	956448877	Auxiliar Administrativo