

INTRODUCCIÓN A LAS BASES DE DATOS RELACIONALES



Ma VICTORIA NEVADO CABELLO



Vision Libros

diseostinta.blogspot.com

**INTRODUCCION
A LAS BASES DE DATOS
RELACIONALES**

M^a Victoria Nevado Cabello

INTRODUCCION
A LAS BASES DE DATOS
RELACIONALES



© INTRODUCCION A LAS BASES DE DATOS RELACIONALES

© M^a Victoria Nevado Cabello

© Editorial: Vision Libros

Calle San Benito 21 Local

Tel: 0034 91 3117696 url: www.visionlibros.com

Vision Libros es marca registrada de Vision Netware S.L.

ISBN: 978-84-9886-809-8

Depósito legal: SE-561-2010

Printed by Publidisa

Diseño base cubierta: Marian González,
reflejandoelazul@visionnet.es

Maquetación: Alvaro Llorente Ayuso

Distribuye y produce:

Distribuidora de publicaciones C/ Magnolias 35 Bis,

Local 28029 Madrid.

Pedidos a pedidos@visionnet.es

www.distribuciondepublicaciones.com

Reservados todos los derechos. Esta publicación no puede ser reproducida, ni registrada, sin el permiso previo por escrito de los titulares del copyright.

ÍNDICE

ÍNDICE.....	7
PRÓLOGO.....	9
INTRODUCCIÓN.....	11
1.- MODELOS DE DATOS	13
1.1.- Introducción a los Modelos de Datos.....	13
1.2.- Los Modelos de Datos y sus propiedades	14
2.- LAS BASES DE DATOS.....	17
2.1.- Necesidad de gestionar los datos	17
2.2.- Sistema de Información	17
2.3.- Tipos de Sistemas de Información.....	18
2.4.- Bases de Datos. Ventajas e inconvenientes	20
2.5.- Características principales de las Bases de Datos.....	21
2.6.- Componentes de una Base de Datos	22
2.7.- Objetivos de los Sistemas Gestores de Base de Datos	25
2.8.- Visiones de los datos y niveles de abstracción.....	26
2.9.- Transformación entre esquemas.....	28
2.10.- El Diccionario de Datos.....	29
2.11.- Forma de trabajo de un SGBD.....	30
2.12.- Sistemas Gestores de Bases de Datos y las Bases de Datos	32
2.13.- Funciones de un SGBD.....	32
2.14.- Componentes de un SGBD	33
2.15.- Lenguaje de Definición de Datos (DDL).....	36
2.16.- Lenguaje de Manipulación de Datos (DML).....	37
3.- EL MODELO ENTIDAD-INTERRELACIÓN	39
3.1.- Estática del Modelo Entidad-Interrelación.....	40
3.2.- Semántica de las interrelaciones	48
3.3.- Generalización y Herencia	50
3.4.- Dinámica del Modelo Entidad-Interrelación.....	56

4.- EL MODELO RELACIONAL.....	57
4.1.- Propiedades de las relaciones	58
4.2.- Definición formal de relación	61
4.3.- Restricciones	64
4.4.- Los valores nulos	65
4.5.- Las doce reglas de Codd.....	66
5.- CÓMO PASAR DEL MODELO ENTIDAD-INTERRELACIÓN AL MODELO RELACIONAL	71
5.1.- Pasos iniciales de la traducción.....	71
5.2.- Transformación de los tipos de entidad.....	74
5.3.- Transformación de los tipos de interrelación 1:1	74
5.4.- Transformación de los tipos de interrelación 1:N.....	78
5.5.- Transformación de los tipos de interrelaciones N:M	79
5.6.- Transformación de los tipos de interrelaciones n-arias	80
5.7.- Transformación de los tipos de interrelaciones reflexivas	81
5.8.- Eliminación de las Relaciones Jerárquicas	83
6.- NORMALIZACIÓN	89
6.1.- Concepto de dependencia funcional.....	90
6.2.- Dependencia funcional completa.....	91
6.3.- Dependencia Funcional Multivaluada (DMV)	93
6.4.- Primera Forma Normal	94
6.5.- Segunda Forma Normal	95
6.6.- Tercera Forma Normal.....	97
6.7.- Forma Normal de Boyce-Codd (FNBC).....	98
6.8.- Cuarta Forma Normal.....	100
6.9.- Quinta Forma Normal.....	103
BIBLIOGRAFÍA	105

PRÓLOGO

Este libro lleva como título “Introducción a las Bases de Datos Relacionales”, y va por tanto orientado a aquellas personas que no poseen conocimientos de las bases de datos relacionales y desean iniciarse en ellas. También puede ser muy útil para las personas que aún teniendo conocimientos de estas bases de datos desean una mayor profundización en ellas.

Dado que las bases de datos de tipo relacional son las que más se utilizan hoy en día, este libro pretende que el lector pueda conocer y poder desarrollar su trabajo con estas bases de datos.

En este libro se pretende que el lector no sólo adquiera conocimientos teóricos, sino también conocimientos prácticos, para que pueda aplicar los conocimientos adquiridos a su tarea mediante estas bases de datos.

INTRODUCCIÓN

El desarrollo tecnológico de nuestra sociedad ha traído la informatización de todas las empresas, negocios, educación, la administración pública, transportes, etc. Ningún área de la sociedad se ha quedado sin informatizar. Esto ha supuesto tener que crear programas informáticos para realizar las tareas que antes se hacían manualmente. Estos programas manejaban datos de entrada y producían datos en su salida. Pero los datos debían de almacenarse de alguna manera para que los programas pudieran acceder a ellos y manejarlos sin provocar errores. Además en algunos sistemas se hacía necesario que varias personas accedieran simultáneamente a la misma información y la modificaran. Todas estas necesidades hicieron que algunas personas y empresas comenzaran a investigar cómo poder solventar todos estos problemas. Se crearon modelos de datos que permitían representar la información real que debía ser informatizada. Una de las primeras empresas que creó un modelo de datos fue IBM, esta empresa creó el Modelo Jerárquico, llamada así porque la información se organizaba como una jerarquía con estructura de árbol. La relación entre las entidades de este modelo siempre era del tipo padre/hijo, donde un hijo sólo podía tener un padre. Después surgió el modelo Codasyl o modelo en red que se asemejaba bastante al modelo jerárquico, pero aquí en registro puede tener varios padres.

Posteriormente Codd en 1970 desarrolló el modelo Relacional, en este modelo los datos se organizan en tablas cuyos datos se relacionan. Este modelo es el más utilizado actualmente y es el modelo que se desarrolla en este libro.

Este libro tiene como título “Introducción a las bases de datos relacionales”, es por tanto un libro que permite al lector iniciarse en el conocimiento de las bases de datos, en la necesidad de la utilización de las bases de datos y en sus características principales.

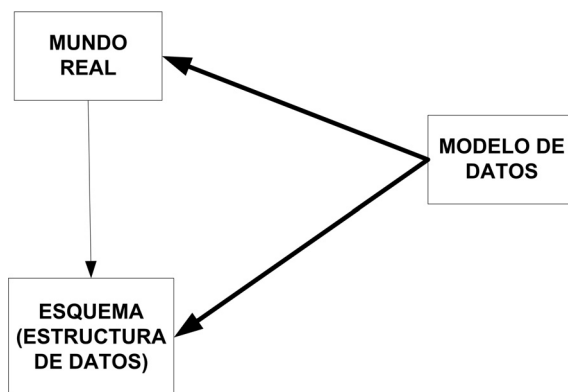
Este libro ha sido desarrollado pensando en aquellas personas que no poseen conocimientos de las bases de datos, y también permite a aquellas personas con conocimientos de ellas, ampliar su conocimiento en las bases de datos, centrándose especialmente en las bases de datos relacionales.

Además de los conocimientos teóricos consta de ejemplos explicativos y de gráficos, que permiten al lector una mejor comprensión y claridad.

I.- MODELOS DE DATOS

I.1.- Introducción a los Modelos de Datos

El ser humano para resolver sus problemas ha tenido que analizar la parcela del mundo real que desea resolver. Dado que el mundo real es demasiado grande para analizarlo, el hombre ha tenido que tomar sólo la parte o parcela de dicho mundo que le interesa. A esa parte del mundo real es lo que llamaremos *Universo del Discurso*. El hombre no sólo debe analizar la realidad, sino que también debe de abstraer lo que desea de esa parcela de realidad. El proceso de abstracción permite interpretar, simplificar y reducir los parámetros y las relaciones del mundo real. Al interpretar la realidad lo que hacemos realmente es indicar las propiedades que caracterizan esa realidad y establecer un *modelo inicial* que permita representarlo.



El *proceso de abstracción* busca las propiedades de un conjunto de objetos reduciendo la complejidad y ayudando a entender el mundo real.

Modelar es una labor intelectual mediante la cual representamos la realidad y en pasos sucesivos llegamos a una estructura adecuada para almacenar datos.

El modelo establece para cada propiedad, que hemos obtenido en el proceso de abstracción de la realidad, cómo se medirá dicha propiedad y las relaciones entre todas las propiedades obtenidas.

Denominamos *datos* al conjunto de propiedades que caracterizan a la realidad que deseamos estudiar o fenómeno.

Denominamos *información* al conjunto de valores que toman estas propiedades que pueden representar un determinado fenómeno.

Los modelos ofrecen distintas formas de abstracción:

- Clasificación / Particularización.
- Agregación / Desagregación.
- Generalización / Especialización
- Asociación / disociación
- Jerarquía de Abstracciones.

1.2.- Los Modelos de Datos y sus propiedades

El Modelo de Datos (MD) se puede definir como un conjunto de conceptos, reglas y convenciones que permiten describir los datos de una parcela del mundo real, llamada Universo del Discurso.

También podemos definir Modelo de Datos como abstracciones mediante las cuales puede realizarse una representación de los problemas que se desea resolver. Es la estructura de una base de datos a la cual se denomina *Esquema*.

Las propiedades del Universo del Discurso son de dos tipos:

- **Estáticas:** Son las propiedades relativamente invariantes en el tiempo(vienen descritas en el MD por las estructuras)
- **Dinámicas:** Son las propiedades que varían en el transcurso del tiempo (en el MD son los operadores).

El Modelo de Datos debe captar tanto las propiedades estáticas como las dinámicas.

El Modelo de Datos no puede describir simultáneamente la naturaleza estática y dinámica de un sistema, para ello precisa dos submodelos:

- **Submodelo Estático:** Representa la estructura del sistema que se desea representar. Consta de:
 - **Elementos permitidos** son:
 - los objetos o entidades
 - las interrelaciones entre estos objetos
 - las propiedades de los objetos llamadas atributos.
 - Los conjuntos de valores sobre los que se definen las propiedades, también llamados dominios.
- **Restricciones**, que son de dos tipos:
 - **Inherentes al modelo**, es la limitación que impone el mismo modelo.
 - **Restricciones semánticas o de integridad**, viene impuesta por la naturaleza del problema. Las restricciones semánticas se deducen de supuestos semánticos explícitos o implícitos o derivados de nuestro conocimiento del mundo real
 - **Submodelo dinámico:** Corresponde a los valores que toman los objetos de un esquema en un momento determinado. A estos valores se le llama ocurrencia del esquema.

La semántica del problema debe reflejarse en la base de datos. Para ello los modelos de datos han de permitir representar las restricciones de integridad.

2.- LAS BASES DE DATOS.

2.1.- Necesidad de gestionar los datos

En el mundo actual existe una cada vez mayor demanda de datos. Esta demanda siempre ha sido patente en empresas y sociedades, pero en estos años la demanda todavía de ha disparado más debido al acceso multitudinario a las redes integradas en Internet y a la aparición de pequeños dispositivos (móviles y PDA) que también requieren esa información.

Desde su nacimiento, la informática se ha encargado de proporcionar herramientas que faciliten la gestión de los datos. Antes de la aparición de las aplicaciones informáticas, las empresas tenían como únicas herramientas de gestión de datos a los cajones, carpetas y fichas en las que se almacenaban los datos. En este proceso manual, el tipo requerido para manipular estos datos era enorme. Sin embargo el proceso de aprendizaje era relativamente sencillo ya que se usaban elementos que el usuario reconocía perfectamente.

Por esa razón, la informática ha adaptado sus herramientas para que los elementos que el usuario maneja en el ordenador se parezcan a los que utilizaba manualmente. Por eso en informática se sigue hablando de ficheros, formularios, carpetas, directorios, etc.

2.2.- Sistema de Información

Un Sistema de Información (SI) es un conjunto formal de procesos, que recopilan , elaboran y distribuyen la información o parte de ella, para poder realizar la operaciones propias de

una empresa, sus actividades de control y dirección según su estrategia de negocio.

Un Sistema de Información consta de los siguientes componentes:

- **Datos.** Se trata de la información relevante que almacena y gestiona el sistema de información. Los datos por sí solos no aportan conocimiento, es necesario procesarlos y transformarlos. La información es el resultado de esta transformación.
- **Hardware.** Equipamiento físico que se utiliza para gestionar los datos. Permite la comunicación, el procesamiento y el almacenamiento de la información.
- **Software.** Aplicaciones que permiten el funcionamiento adecuado del sistema
- **Recursos humanos.** Personal que maneja el sistema de información. Introducen, manejan o usan la información para realizar sus actividades en función de los procedimientos de trabajo establecido.

2.3.- Tipos de Sistemas de Información

Existen dos tipos de sistemas de información:

- Sistema de información orientados al proceso
- Sistema de información orientados a los datos

Sistemas de información orientados al proceso

En estos sistemas de información se crean diversas aplicaciones (software) para gestionar diferentes aspectos del sistema. Cada aplicación realiza unas determinadas operaciones. Los datos de dichas aplicaciones se almacenan unidades de almacenamiento del ordenador (archivos binarios, archivos de texto o en hojas de cálculo).

Cada programa almacena y utiliza sus propios datos. La única ventaja de este sistema, es que los procesos eran inde-

pendientes, por lo que la modificación de uno no afectaba al resto. Pero tiene una serie de inconvenientes:

- **Datos redundantes.** Ya que se repiten continuamente.
- **Datos inconsistentes.** Ya que un proceso cambia sus datos y no el resto. Por lo que el mismo dato puede tener valores distintos según qué aplicación acceda a él.
- **Coste de almacenamiento elevado.** Al almacenarse varias veces el mismo dato, se requiere más espacio en los discos.
- **Difícil acceso a los datos.** Cada vez que se requiera una consulta no prevista inicialmente, hay que modificar el código de las aplicaciones o incluso crear una nueva aplicación.
- **Dependencia de los datos a nivel físico.** Para poder saber cómo se almacenan los datos, es decir qué estructura se utiliza de los mismos, necesitamos ver el código de la aplicación; es decir el código y los datos no son independientes.
- **Tiempos de procesamiento elevados.** Al no poder optimizar el espacio de almacenamiento.
- **Dificultad para el acceso simultáneo a los datos.** Es casi imposible de conseguir ya que se utilizan archivos que no admiten esta posibilidad. Dos usuarios no pueden acceder a los datos de forma concurrente.
- **Dificultad para administrar seguridad.** Cada aplicación se crea independientemente; es por tanto muy difícil establecer criterios de seguridad uniformes.

A estos sistemas se les llama *sistemas de gestión de ficheros*. Se consideran también así a los sistemas que utilizan programas ofimáticos (como **Word** y **Excel**) para gestionar sus datos.

Sistemas de información orientados a los datos. Bases de Datos

En este tipo de sistemas los datos se centralizan en una **base de datos** común a todas las aplicaciones. En esos sistemas los datos se almacenan en una única estructura lógica que es uti-

lizable por diferentes aplicaciones. A través de esa estructura se accede a los datos que son comunes a todas las aplicaciones.

2.4.- Bases de Datos. Ventajas e inconvenientes

Las bases de datos permiten mejorar la calidad de las prestaciones de los sistemas informáticos y aumentar su rendimiento.

Las bases de datos tienen las siguientes *ventajas*:

- **Independencia de los datos y los programas y procesos.** Esto permite modificar los datos sin modificar el código de las aplicaciones.
- **Menor redundancia.** No hace falta tanta repetición de datos. Sólo se indica la forma en la que se relacionan los datos.
- **Integridad de los datos.** Mayor dificultad de perder los datos o de realizar incoherencias con ellos.
- **Coherencia de los resultados.** Al recogerse y almacenarse la información una sola vez, en los tratamientos se utilizan siempre los mismos datos, por lo que los resultados son coherentes.
- **Mayor seguridad en los datos.** Al permitir limitar el acceso a los usuarios. Cada tipo de usuario podrá acceder a unas cosas.
- **Datos más documentados.** Gracias a los **metadatos** que permiten describir la información de la base de datos.
- **Acceso a los datos más eficiente.** La organización de los datos produce un resultado más óptimo en rendimiento.
- **Reducción del espacio de almacenamiento.** Gracias a una mejor estructuración de los datos.
- **Acceso simultáneo a los datos.** Es más fácil controlar el acceso de usuarios de forma concurrente.

Las bases de datos también tienen una serie de *desventajas*:

- **Instalación costosa.** El control y administración de bases de datos requiere de un software y hardware de elevado

coste. Además de la adquisición y mantenimiento del Sistema Gestor de Base de Datos (SGBD).

- **Requiere personal cualificado.** Debido a la dificultad de manejo de este tipo de sistemas.
- **Implantación larga y difícil.** Debido a los puntos anteriores. La adaptación del personal es mucho más complicada y lleva bastante tiempo.
- **Ausencia de estándares reales.** Lo cual significa una excesiva dependencia hacia los sistemas comerciales del mercado. Aunque, hoy en día, una buena parte de esta tecnología está aceptada como estándar de hecho.
- **Falta de rentabilidad a corto plazo.** Debido al coste de personal y quipos, además del tiempo que tarda en estar operativa.

2.5.- Características principales de las Bases de Datos

Las características que debe tener una Base de Datos son:

- **Independencia lógica y física de los datos.**
 - **Independencia lógica de los datos**, que permite cambiar la representación lógica del problema sin afectar a los programas de aplicación que los manipulan.
 - **Independencia física de los datos**, que permite que la distribución en las unidades de almacenamiento y la estructura física de la información almacenada sea independiente de los cambios de la estructura lógica, y por tanto, de los procedimientos que manejan las mismas.
- **Versatilidad en la representación de la información.-** La organización de la información en la base de datos debe permitir que diferentes procedimientos puedan construir diferentes registros a partir de la información existente de la base de la base de datos.
- **Mínima redundancia.-** La redundancia provoca problemas de inconsistencia de la información y necesidades de

almacenamiento mayores. Las bases de datos surgieron para evitar esta redundancia, pero siempre existe algo de redundancia, lo que denominamos *redundancia mínima*.

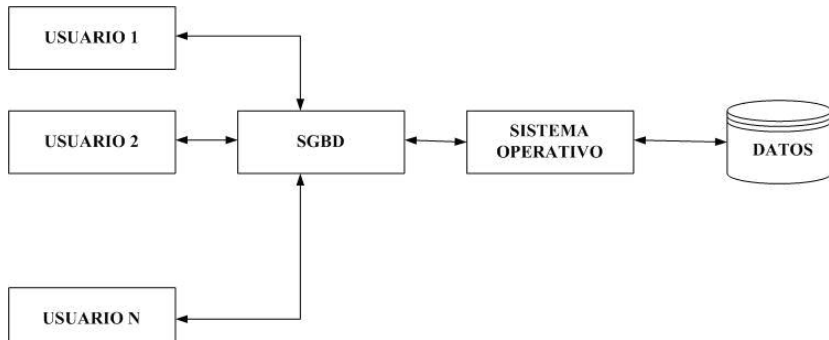
- **Simplicidad.**- Las bases de datos deben estar basadas en representaciones lógicas simples, para permitir la verificación en la representación del problema que representan.
- **Capacidad de proceso.**- Una base de datos debe responder en un tiempo aceptable, a cualquier consulta a la base de datos.
- **Integridad.**- Es la veracidad de los datos almacenados respecto a la información existente en el dominio del problema que trata la base de datos. Los datos no deben ser modificados o destruidos de forma anómala. Los procedimientos de inserción o actualización deben asegurar que el sistema pueda garantizar la integridad de la información.
- **Privacidad.**- Es evitar que personas no autorizadas puedan acceder a la información de la base de datos.
- **Seguridad.**- Permite proteger los datos frente a la pérdida total o parcial de información, por fallos del sistema o por accesos accidentales o intencionados de los mismos. La base de datos debe satisfacer los requisitos siguientes:
 - Seguridad contra la destrucción de los datos causados por el entorno: fuego, robo, inundaciones, etc.
 - Seguridad contra destrucción de los datos causada por fallos del sistema (hardware o software) de forma que los datos puedan reconstruirse.
 - Seguridad contra accesos no autorizados a la base de datos.
 - Seguridad contra accesos indebidos a los datos.

2.6.- Componentes de una Base de Datos

Una base de datos es una colección interrelacionada de datos, almacenados en un conjunto sin redundancias innecesarias cuya finalidad es la de servir a una o más aplicaciones de la manera más eficiente.

Los componentes de una base de datos son:

- **Los datos:** El componente fundamental de una base de datos son los datos que están interrelacionados entre sí, formando un conjunto con un mínimo de redundancias.
- **El software:** Los datos, para que puedan ser utilizados por diferentes usuarios y diferentes aplicaciones, deben estar estructurados y almacenados de forma independiente de las aplicaciones. Para ello se utiliza un software o conjunto de programas que actúa de interfaz entre los datos y las aplicaciones. A este software se le denomina Sistema de Gestión de Base de Datos (SGBD). El SGBD crea y organiza la base de datos, y además atiende todas las solicitudes de acceso hechas a la base de datos tanto por los usuarios como por las aplicaciones.



· **Recursos Humanos:**

- **Informáticos:** Son los profesionales que definen y preparan la base de datos. Pueden ser:

- **Directivos/as.** Son los organizadores y coordinadores del proyecto a desarrollar. Esto significa que son los encargados de decidir los recursos que se pueden utilizar, planifican el tiempo y las tareas, dirige las entrevistas y reuniones pertinentes.
 - **Analistas.** Son los encargados de controlar el desarrollo de la base de datos aprobada por la dirección. Diseñan de la base de datos (especialmente de los esquemas interno y conceptual) y los coordinadores de la programación de la misma.
 - **Administradores/as de las bases de datos.** Definen la seguridad de la base de datos y gestionan las copias de seguridad y la gestión física de la base de datos. Los analistas suelen tener esta funcionalidad cuando la base de datos está creada.
 - **Desarrolladores/as o programadores/as.** Encargados de la realización de las aplicaciones de usuario de la base de datos.
 - **Equipo de mantenimiento.** Encargados de dar soporte a los usuarios en el trabajo diario (suelen incorporar además tareas administrativas).
- **Usuarios:** Pueden ser de dos tipos:
- **Expertos/as.** Utilizan el lenguaje de manipulación de datos (DML) para acceder a la base de datos. Son usuarios que utilizan la base de datos para gestión avanzada de decisiones.
 - **Habituales.** Utilizan las aplicaciones creadas por los desarrolladores para consultar y actualizar los datos. Son los que trabajan en la empresa a diario con estas herramientas.
 - **Ocasionales.** Son usuarios que utilizan un acceso mínimo a la base de datos a través de una aplicación que permite consultar ciertos datos.

2.7.- Objetivos de los Sistemas Gestores de Base de Datos

Un sistema gestor de bases de datos o **SGBD** (aunque se suele utilizar más a menudo las siglas **DBMS** procedentes del inglés, *Data Base Management System*) es el software que permite a los usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos.

En estos Sistemas se proporciona un conjunto coordinado de programas, procedimientos y lenguajes que permiten a los distintos usuarios realizar sus tareas habituales con los datos, garantizando además la seguridad de los mismos.

El éxito del SGBD reside en mantener la seguridad e integridad de los datos. Lógicamente tiene que proporcionar herramientas a los distintos usuarios. Entre las herramientas que proporciona están:

- **Herramientas para la creación y especificación de los datos.** Así como la estructura de la base de datos.
- **Herramientas para administrar y crear la estructura física** requerida en las unidades de almacenamiento.
- **Herramientas para la manipulación de los datos** de las bases de datos. para añadir, modificar, suprimir o consultar datos.
- **Herramientas de recuperación** en caso de desastre.
- **Herramientas para la creación de copias de seguridad.**
- **Herramientas para la gestión de la comunicación** de la base de datos.
- **Herramientas para la creación de aplicaciones** que utilicen esquemas externos de los datos.
- **Herramientas de instalación** de la base de datos.
- **Herramientas para la exportación e importación de datos.**

2.8.-Visiones de los datos y niveles de abstracción

Dependiendo de quién acceda o use la base de datos pueden existir tres tipos de visiones de los datos en una base de datos, y son las siguientes:

- **Visión externa.**-Es la visión de los datos que tienen los usuarios finales de una base de datos. Un usuario trata sólo una visión parcial de la información, sólo aquella que interviene en el dominio de su actividad. Se corresponde con los *Esquemas o Subesquemas Externos* que tiene cada usuario de la base de datos.
- **Visión conceptual.**- Es la representación abstracta del problema. Una base de datos representa la información de un problema del mundo real. Es independiente de:
 - Cómo va a ser tratada esta información.
 - Las visiones externas que tenga.
 - Cómo va a almacenarse esta información físicamente.

Esta visión no va a cambiar a no ser que cambie la naturaleza del problema. El diseñador o analista es el encargado de describir esta visión. Se corresponde con el *Esquema Conceptual*

- **Visión física.**- Es la representación de cómo la información es almacenada en los dispositivos de almacenamiento. Describe las estructuras u organizaciones físicas, dispositivos, ficheros, tipos de datos, etc. El administrador de la base de datos es el encargado de definir esta visión física de los datos. Se corresponde con el *Esquema interno o físico*.

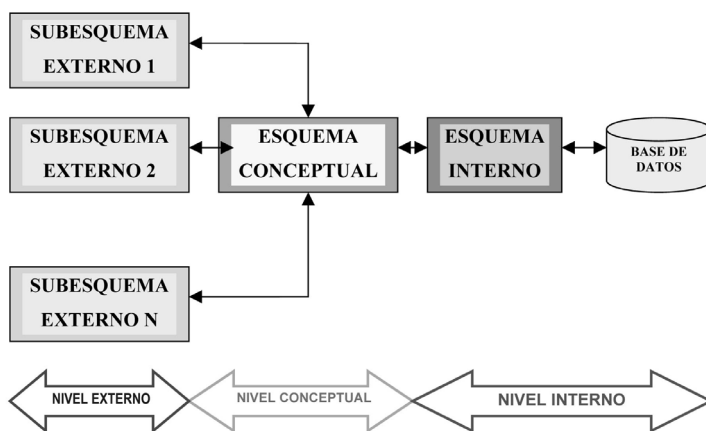
Estas tres visiones se corresponden con los tres niveles de abstracción. Estos tres niveles de abstracción garantizan la independencia de los datos:

- Poder modificar la organización física sin que haya cambiado la descripción conceptual, y sin modificar los programas de aplicación que manipulan esa información.

- Poder modificar la representación conceptual sin que haya que modificar la estructura física de la información, ni los programas de aplicación, siempre y cuando no se eliminen de la representación conceptual objetos necesarios o requeridos en estos otros niveles.
- Poder cambiar las visiones externas conforme a nuevos requerimientos funcionales sin que por ello se tenga que modificar ninguna descripción de datos a ningún nivel de abstracción.

El nivel más importante es el nivel conceptual, ya que los otros dos niveles se apoyan en éste. Una descripción conceptual incluirá todas las entidades que intervienen en el problema, sus atributos y propiedades, así como las relaciones existentes entre ellas.

Existen en el mercado diferentes paquetes de SCBD con diferentes arquitecturas, la más estandarizada es la normativa ANSI/X3/SPARC que podemos ver en el siguiente gráfico, en él podemos observar los tres niveles de abstracción y los tres tipos de esquemas.

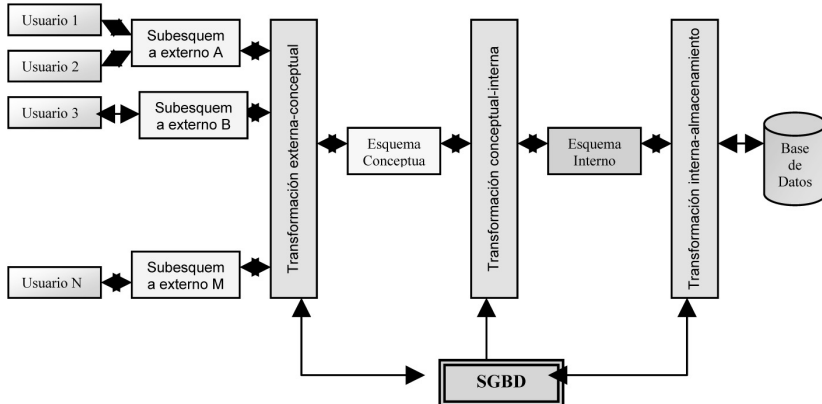


2.9.- Transformación entre esquemas

En un sistema que tenga los tres niveles de abstracción, el SGBD debe garantizar la transferencia de los datos desde el formato de un nivel a otro. Este proceso se llama transformación de los datos o *mapeo*.

Existen tres niveles de transformación:

- **Transformación Interno/Almacenamiento:** permite pasar del esquema interno a los datos físicos de la base de datos y viceversa.
- **Transformación Conceptual / Interno:** permite pasar del esquema conceptual al esquema interno y viceversa
- **Transformación Externo / Conceptual:** permite pasar del subesquema externo al esquema conceptual y viceversa.



El SGBD debe conocer las relaciones existentes entre los niveles para así poder efectuar el mapeo de los datos de un nivel a otro. Un esquema se traduce en otro utilizando unas *reglas de correspondencia*, que son los parámetros que definen los procedimientos de mapeo de los datos de un esquema a otro. Cuando se solicita un acceso a la base de datos el SGBD realiza los siguientes pasos:

1. Interpreta la solicitud.
2. Inspecciona:
 - a) El Esquema Externo.
 - b) La transformación Externa/Conceptual.
 - c) El Esquema Conceptual.
 - d) La transformación Conceptual/Interna.
 - e) El Esquema Interno.
 - f) La transformación Interna/Almacenamiento.
3. Realiza la operación sobre el dato físico.

2.10.- El Diccionario de Datos

Todos estos datos son construidos y mantenidos por el Administrador de la Base de Datos. Una de las herramientas más importantes con que cuenta el Administrador de la Base de datos para el manejo de los datos es el *Diccionario de Datos*.

En el Diccionario de Datos se almacenan:

- Las descripciones interna, conceptual y externa de la base de datos, así como las reglas de correspondencia necesarias para pasar de un esquema a otro.
- Los descriptores de los campos, registros y referencias cruzadas entre los registros de varios ficheros.
- Los códigos de autorización y seguridad de los datos.
- Los esquemas externos que emplea cada aplicación, quiénes son sus usuarios y qué autorizaciones tienen.

2.11.- Forma de trabajo de un SGBD

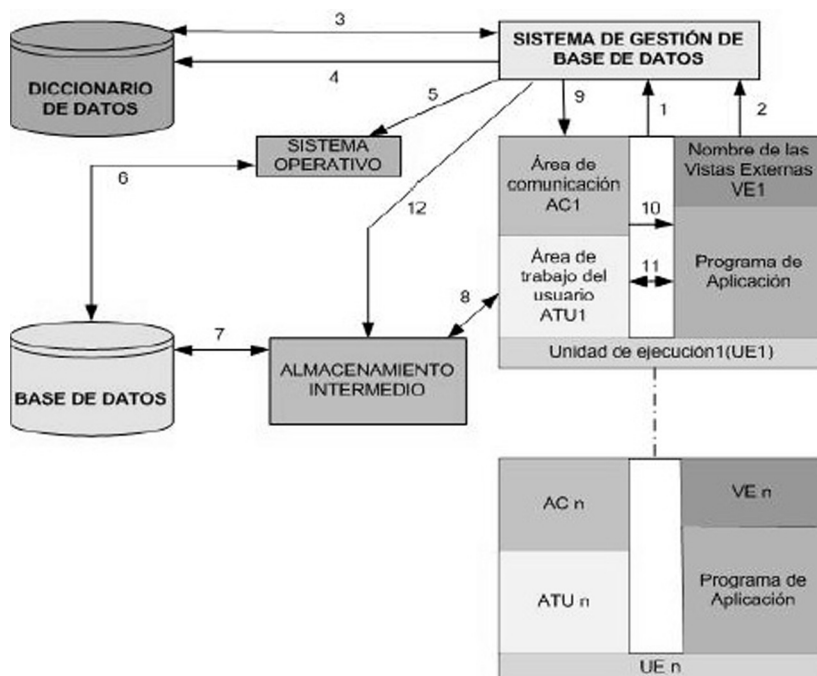
El SGBD es un subsistema del Sistema Informático. El SGBD está interrelacionado con el Sistema Operativo.

Cada programa de aplicación que se está ejecutando existe una unidad de ejecución (UE), donde se encuentra un área de trabajo de usuario (ATU), con sus áreas de entrada/salida, y un área de comunicación con el SGBD destinada a recibir los mensajes y la información de control procedente de éste. Un programa de aplicación realiza una referencia a una Vista Externa definida para ese programa y guardada en el Diccionario de Datos, junto con los Esquema Lógico e Interno.

Veamos los pasos del funcionamiento del SGBD:

- Paso 1.-** Desde la UE se realiza una llamada al SGBD (1) haciendo referencia a la Vista Externa implicada (2).
- Paso 2.-** El SGBD analiza la llamada y completa los argumentos con la información de la vista a la que se ha hecho referencia, así como con el correspondiente esquema lógico y estructura interna. Esta información pasa del Diccionario de Datos al SGBD (3 y 4).
- Paso 3.-** Se comprueba si el programa tiene derecho a utilizar esa vista. El SGBD traduce la llamada en las correspondientes órdenes para los métodos de acceso al sistema operativo (5). Si el sistema había utilizado recientemente los mismos datos, es posible que éstos todavía estén almacenados en el área de almacenamiento intermedio (12).
- Paso 4.-** El Sistema Operativo accede al disco donde se encuentran almacenado los datos (6).
- Paso 5.-** Los datos recuperados pasan del soporte al área de almacenamiento intermedio (buffer). Si se tratase de una inserción o modificación pasarían en sentido contrario (7).

- Paso 6.-** Se produce la transferencia de los datos desde el buffer al área de trabajo del usuario de la UE que hizo la llamada (8), o en sentido contrario si se trata de una inserción o modificación.
- Paso 7.-** El SGBD pasa los indicadores de estado al área de comunicación (9), indicándose así si la operación se ha realizado correctamente o no.
- Paso 8.-** El programa revisa el estado de los indicadores del área de control, y toma las decisiones oportunas (10).
- Paso 9.-** Si la operación se ha realizado correctamente, los datos que se encuentran en el área de trabajo pueden ser utilizados por el programa.



2.12.- Sistemas Gestores de Bases de Datos y las Bases de Datos

No debemos confundir una Base de Datos con un Sistema Gestor de Base de Datos. Una Base de datos es la información almacenada, que cumple una serie de características y restricciones, pero para que es la información pueda ser almacenada y el acceso a la misma satisfaga las características exigidas a una base de datos, es necesario que exista una serie de procedimientos, un sistema software, que sea capaz de llevar a cabo tal labor. A este sistema software es lo que llamamos Sistema Gestor de de Base Datos (SGBD).

Un SGBD permite:

- Definir los datos a los distintos niveles de abstracción (físico, lógico y externo).
- Manipulación de los datos en la base de datos. Permitiendo insertar, modificar, borrar y consultar los datos.
- Mantenimiento de la integridad de la base de datos.
- Control de la privacidad y seguridad de los datos en la base de datos.

2.13.- Funciones de un SGBD

Las funciones principales de un SGBD son las de descripción, manipulación y control.

Función de definición

Va a permitir al diseñador de la base de datos especificar los elementos que la integran, su estructura y las relaciones que existen entre ellos, las reglas de integridad semántica, etc., así como las características de tipo físico y las vistas lógicas de los usuarios.

Esta función la realiza el Lenguaje de Definición de Datos (DDL), que es propio de cada SGBD.

Función de manipulación

Después de describir los datos, es necesario cargarlos en las estructuras previamente creadas, para posteriormente poder utilizarlos. Los usuarios podrán recuperar la información o actualizarla. La consulta podrá ser de dos tipos:

- Consulta selectiva.- donde se localizan registros que cumplen una determinada condición, según un criterio de selección.
- Consulta sobre la totalidad de los datos.- donde se recuperan todos los datos de la BD o todos los de un determinado tipo.

La actualización podrá realizarse con tres operaciones diferentes:

- Borrado o eliminación de elementos.
- Modificación de datos.
- Inserción de nuevos elementos.

La función de manipulación se realizará a través del Lenguaje de Manipulación de Datos (DML).

Función de Control

Debe integrar una serie de instrumentos para facilitar la tarea del administrador. Permite funciones de servicio como:

- Cambiar la capacidad de los ficheros.
- Obtener estadísticas de utilización y funciones de seguridad como:
 - Copias de seguridad
 - Rearranque del sistema
 - Protección frente a accesos no autorizados.

2.14.- Componentes de un SGBD

Las funciones anteriores se realizan gracias a una serie de componentes son:

- **DDL o Lenguaje de Definición de Datos.-** Permite definir la representación lógica de los datos que integran la base de datos.
- **DCL o Lenguaje de Control de Datos.-** Permite el control de acceso a la información almacenada en el diccionario de datos.
- **DSDL o Lenguaje de Definición de Almacenamiento de datos.-** En la mayoría de los SGBD el mismo lenguaje DDL permite la definición de los datos en el nivel de representación físico, si bien en otros es un subcomponente de éste denominado lenguaje de definición de almacenamiento de datos. Se definen los datos del dominio de un problema a los dos niveles de abstracción, denominado *Esquema de la Base de Datos*. En el esquema estarán definidas:
 - Las características del problema a nivel lógico: clases de objetos con sus propiedades, relaciones entre objetos y sus propiedades restricciones de las clases de objetos y de las relaciones.
 - Las características del problema a nivel físico: unidades físicas, volúmenes y archivos, características físicas y lógicas de los medios de almacenamiento y métodos de acceso a la información.
- **DML o Lenguaje de Manipulación de Datos.-** Permite realizar dos funciones en la gestión de los datos:
 - Definición del nivel externo o de usuario de los datos.
 - La manipulación de los datos (inserción, borrado, modificación y recuperación).
- **El Gestor de la Base de Datos.-** Es un componente software encargado de garantizar el correcto, eficiente, íntegro y seguro acceso y almacenamiento de los datos. Además

proporciona un interfaz entre los datos almacenados y los programas de aplicación. Se encarga de:

- Garantizar la privacidad de los datos.
 - Garantizar la integridad de los datos, gestionando que los datos que se almacenan en la base de datos satisfagan las restricciones definidas en el esquema de la misma.
 - Garantizar la seguridad de los datos, realizando procedimientos que puedan recuperar datos después de un fallo que ocasione pérdida o deterioro temporal de los mismos.
 - Garantizar el acceso concurrente a la base de datos de forma que varios usuarios puedan acceder al mismo o distinto dato sin que se provoque pérdida en la integridad de los datos.
 - Interaccionar con el sistema operativo, en concreto con el gestor de archivos.
- **El administrador de la base de datos (DBA).**- Es un componente humano con una serie de responsabilidades como la definición, administración, seguridad, privacidad e integridad de la información. Sus tareas son las siguientes:
- Definición del *esquema lógico o conceptual* de la base de datos, mediante sentencias DDL.
 - Definición del *esquema físico* de la base de datos, especificando las estructuras de almacenamiento y métodos de acceso a la información almacenada en los dispositivos físicos.
 - Definición de subesquemas o visiones externas del usuario.
 - Mantenimiento de los esquemas, con modificaciones del esquema lógico, en la representación física de datos o en los subesquemas de usuarios.
 - Control de la privacidad de los datos.

2.15.- Lenguaje de Definición de Datos (DDL)

El DDL permite definir la representación lógica de los datos que integran la base de datos, su estructura y las relaciones que existen entre ellos, las reglas de integridad, los controles de acceso, las características de tipo físico y las vistas externas de los usuarios.

El DDL tiene una función de descripción que se puede ver desde distintos puntos de vista.

- Desde el *punto de vista lógico o conceptual*, el Administrador de la Base de Datos utiliza el DDL para asignar nombres a los campos, agregados de los datos, a los registros, etc., estableciendo longitudes y características, así como las relaciones entre estos elementos y las restricciones semánticas que se deben aplicar. No se debe incluir información referente a la estructura física.
- Desde el *punto de vista interno*, en un SGBD en el cual fuesen totalmente independiente las estructuras físicas y lógica, el SGBD se podría encargar de definir la estructura física, suministrarle informaciones como volúmenes, crecimiento previsto, tipos de registros más accedidos, etc. Pero en la práctica, el rendimiento del sistema aumenta si el Administrador de la Base de Datos especifica características de la estructura física, junto con la descripción lógica. Por ello el objetivo de independencia física no se puede conseguir.

En este nivel se ha de indicar el espacio (volúmenes, cilindros y pistas) reservado para la base, la longitud de los campos, su modo de representación, y también los caminos de acceso, punteros, índices, etc.

- Desde el *punto de vista externo*, el DDL también se encarga de suministrar medios para decir las vistas externas, usando el mismo lenguaje que para el esquema lógico.

Las definiciones de estos esquemas se compilan con un compilador DDL, y las tablas que se obtienen se almacenan en el Diccionario de Datos. En el Diccionario de Datos se almacenan *metadatos*, es decir, datos acerca de los datos, que será consultado por SGBD.

2.16.- Lenguaje de Manipulación de Datos (DML)

Mediante la función de manipulación los usuarios de la base de datos pueden buscar, añadir, borrar o modificar los datos de la base de datos. Esta función se realiza mediante el Lenguaje de Manipulación de Datos (DML).

El usuario gracias a este lenguaje puede referirse a determinados conjuntos de datos que cumplan un criterio de selección, como que un atributo tenga un determinado valor, o que un conjunto de atributos y valores satisfagan una expresión lógica.

Los DML se pueden clasificar en:

- **Huésped:** Va embebido en un lenguaje de programación anfitrión, como COBOL, C, PASCAL, ADA, FORTRAN. La estructura de un lenguaje huésped que interactúa con una base de datos es la siguiente:

SENTENCIAS DECLARATIVAS

.....

LLAMADA A LA VISTA DE USUARIO

.....

Declaración del área de entrada/salida.

Declaración de área de comunicaciones de mensajes.

SENTENCIAS DE PROCESO

..... LLAMADA A LA SENTENCIA DE DML.

Comprobación del contenido del área de mensajes.

En la parte de proceso se hace una llamada a la base de datos utilizando sentencias DML. Después se deberá comprobar el área de mensajes para saber si la sentencia se ha realizado con éxito o no.

- **Autocontenido:** Dan facilidades a los usuarios para extraer e introducir información de la base de datos de forma interactiva, por ejemplo el lenguaje SQL (Structured Query Language o Lenguaje de Consulta Estructurado)

A veces, el mismo DML puede actuar simultáneamente como lenguaje huésped y autocontenido, como ocurre con el SQL.

Hay varias clasificaciones de DML, una clasificación es:

- **Procedimentales.-** Requieren que en las sentencias del lenguaje se especifique qué datos se van a manipular y qué operaciones deben realizarse.
- **No procedimentales.-** Requieren que en las sentencias se especifique sólo qué datos se van a manipular, siendo el propio DML el encargado de determinar los procedimientos más efectivos para ello.

Otra clasificación es:

- **Navegacionales:** Recuperan o actualizan los datos registro a registro. El programador debe indicar el camino a recorrer para llegar al registro buscado. Cada sentencia opera sólo con un registro (lenguaje jerárquico y de red).
- **No navegacionales:** Actúan sobre un conjunto de registros. Una sola sentencia puede recuperar un conjunto de registros (lenguaje SQL).

3.- EL MODELO ENTIDAD-INTERRELACIÓN

Los modelos de datos soportados por los SGBD debido a su bajo nivel de abstracción, no suelen ofrecer los mecanismos suficientes para captar la semántica del mundo real, por lo que surgen modelos conceptuales, más ricos semánticamente. Entre estos modelos de datos semánticos, el modelo Entidad-Interrelación es posiblemente el más utilizado.

El Modelo Entidad-Interrelación (ME-R) propuesto por Peter Chen en 1970 presenta el modelo como una vista unificada de datos. Este modelo se centra en la estructura lógica y abstracta de los datos, como representación del mundo real, independientemente de las características físicas.

Todo modelo de datos tiene básicamente dos tipos de propiedades:

- **Estáticas.**- Describen la estructura del problema y apenas varían en tiempo. Utilizamos el lenguaje de definición de datos (DDL) para describirlas.
- **Dinámicas.**- Permiten el paso de un estado a otro, ejecutando operaciones sobre los datos almacenados en la estructura. Para expresar operaciones se utiliza el Lenguaje de Manipulación de Datos (DML).

El ME-R, tal como fue propuesto por Chen, no consideraba la parte dinámica, definiendo solamente la parte estructural y semántica, y la forma de representación de los datos. Posteriormente, otros autores completan el modelo proponiendo lenguajes de manipulación, que permiten recuperar y actualizar de forma sencilla los datos almacenados en las estructuras.

El modelo de datos propuesto por Codd tenía los siguientes objetivos:

- **Independencia física.**- El modo en el que se almacenan los datos no influye en su manipulación, y por tanto, los usuarios que acceden a esos datos no tendrán que modificar sus programas por cambios en el almacenamiento físico.
- **Independencia lógica.**- Permite que se pueda añadir, modificar o eliminar objetos de la base de datos sin que ello repercuta en los programas y/o usuarios que acceden a vistas (subconjuntos parciales de información).
- **Flexibilidad.**- Permite poder presentar a cada usuario los datos en la forma en que éste prefiera.
- **Uniformidad.**- Las estructuras lógicas de los datos presentan un aspecto uniforme, facilitando la manipulación de la base de datos.
- **Sencillez.**- El modelo de datos relacional es fácil de comprender y de utilizar por parte del usuario final.

3.1.- Estática del Modelo Entidad-Interrelación

Los conceptos básicos de este modelo son los siguientes:

- Las **entidades**
- Las **interrelaciones**
- Los **dominios**

Entidad

Entidad es una persona, lugar, cosa, concepto o suceso, real o abstracto, de interés en el dominio del problema. Es el objeto del cual deseamos almacenar información en la base de datos.

Se denomina **tipo de entidad** a la estructura genérica y **ocurrencia de la entidad** a cada una de las realizaciones concretas de este tipo de entidad.

La representación gráfica de un tipo de entidad es un rectángulo etiquetado con el nombre del tipo de entidad, tal y como se indica a continuación:

NOMBRE DEL TIPO DE ENTIDAD

Veamos un ejemplo: el tipo de entidad PERSONA se refiere a la estructura que nos describe las características de las personas como una abstracción, mientras que una ocurrencia de PERSONA será cada uno de las personas en concreto.

PERSONA

Existen dos clases de entidades:

- **Regular o Fuerte.**- Las ocurrencias de un tipo de entidad regular tienen existencia propia. Se representa con un rectángulo como se ha visto anteriormente.
- **Débiles.**- La existencia de cada ocurrencia de un tipo de entidad débil depende de la existencia de la ocurrencia del tipo de entidad regular de la que depende. Si se elimina una ocurrencia de un tipo de entidad regular, desaparecen también con ella todas las ocurrencias de la entidad débil dependientes de la misma. Un tipo de entidad débil se representa con dos rectángulos concéntricos con su nombre en el interior. Como ejemplo de tipo de entidad débil tenemos EJEMPLAR de un cierto libro, cuya existencia no tiene sentido, sino existe el correspondiente libro en la base de datos.

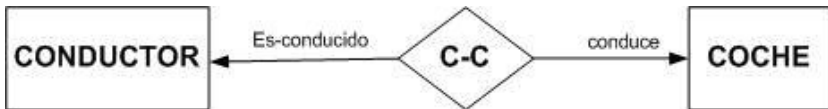
EJEMPLAR

Interrelación

Es la asociación entre relaciones. Denominamos **Tipo de Interrelación** a la estructura genérica del conjunto de interrelaciones existentes entre dos o más tipos de entidad. Llamamos **Ocurrencia de una Interrelación** a la vinculación que existe entre las ocurrencias concretas de cada uno de los tipos de entidad que participan en la interrelación.

La interrelación se representa mediante un rombo etiquetado con el nombre de la interrelación, unido mediante arcos a los tipos de entidad que asocia.

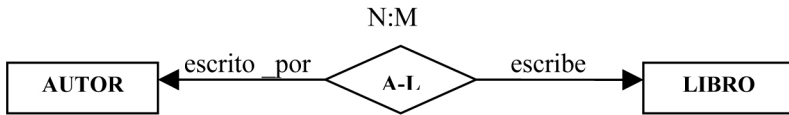
A continuación vemos un ejemplo de tipo de interrelación (C-C) entre dos tipos de entidad (CONDUCTOR y COCHE):



Un tipo de interrelación se caracteriza por:

- **Nombre.-** Por el que identificamos de forma única el tipo de interrelación. Lo escribimos dentro del rombo.
- **Grado.-** Número de tipos de entidad que participan en un tipo de interrelación. Puede ser de grado dos (binarias) cuando asocia dos tipos de entidad, de grado tres (ternarias) cuando asocian tres tipos de entidad, o en general de grado n (n-arias). En el ejemplo anterior el grado de la interrelación **C-C** es de grado dos.
- **Tipo de correspondencia.-** Es el número máximo de ocurrencias de un tipo de entidad que pueden intervenir por cada ocurrencia del otro tipo de entidad asociado a la interrelación. Existen diferentes tipos de correspondencias:

- **Tipo de correspondencia 1:1.**- Cuando en la interrelación sólo pueden aparecer, como máximo una ocurrencia de un tipo de entidad relacionada con una ocurrencia del otro tipo de entidad.
- **Tipo de correspondencia 1:N .**- Cuando una ocurrencia de un tipo de entidad se puede relacionar con un número indefinido de ocurrencias (mayor que uno) del otro tipo de entidad que participa en la interrelación.
- **Tipo de correspondencia N:M.**- Cuando ocurre lo anteriormente descrito, pero para ambos tipos de entidad. Para representarlo gráficamente, se añade una etiqueta al lado del rombo indicando el tipo de interrelación.

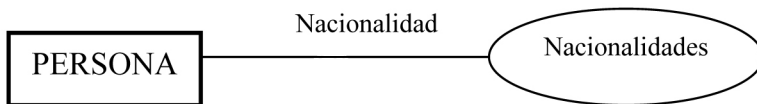


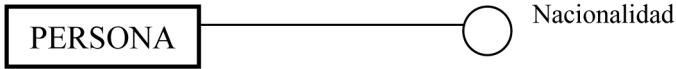
- **Atributo.**- Es cada una de las propiedades características que tiene un tipo de entidad o de interrelación. El tipo de entidad AUTOR tiene como atributos *Nombre*, *Nacionalidad*, *Fec-Nac*, etc. El tipo de entidad LIBRO tiene como atributos *Título*, *ISBN*, etc. El tipo de interrelación A-L tiene como atributo *Orden-de-Firma*. Existen cuatro tipos de atributos:
 - **Obligatorios.**- aquellos que pueden tomar un valor y no se permite ningún ejemplar que no tenga un valor determinado en el atributo.
 - **Opcional.**- aquellos atributos que pueden tener valores o no tenerlos.
 - **Monoevaluado.**- aquel atributo que sólo puede tener un único valor.

- **Multivaluado.**- aquel atributo que puede tener varios valores.

Denominamos **Dominio** al conjunto de posibles valores que puede tomar un atributo. Cada dominio tiene un nombre y una existencia propia con independencia de cualquier entidad o atributo. Como un dominio restringe los valores del atributo, puede ser considerado como una restricción. Matemáticamente, atribuir un dominio a un atributo significa "todos los valores de este atributo deben de ser elementos del conjunto especificado". Por ejemplo el dominio *Nacionalidades* tiene como valores: *española, francesa, italiana*. El atributo *Nacionalidad* de PERSONA estará definido sobre este dominio. La existencia del atributo *Nacionalidad* va unida a la existencia del tipo de entidad PERSONA, mientras que el dominio *Nacionalidades* existe por sí mismo, independiente de cualquier tipo de entidad.

El dominio se representa con un círculo y óvalo en cuyo interior aparece su nombre. El nombre del atributo se escribe sobre el arco que une el dominio con el tipo de entidad o interrelación a la que pertenece dicho atributo. Cuando coincide el nombre del dominio con el del atributo se suele escribir el nombre del atributo en el interior del círculo y óvalo, eliminando el nombre del arco, escribiendo el nombre del atributo al lado del círculo en lugar de en el interior.





Denominamos **atributo identificador principal (AIP)**, al atributo elegido entre todos los atributos de un tipo de entidad que permiten identificar unívoca y mínimamente cada una de las ocurrencias de ese tipo de entidad. Si existe más de un atributo que cumpla esa condición, a esos atributos los denominaremos **atributo identificador candidato (AIC)**, de los cuales se elige uno como principal y los otros son alternativos (**atributo identificador alternativo AIA**).

Se representa mediante:

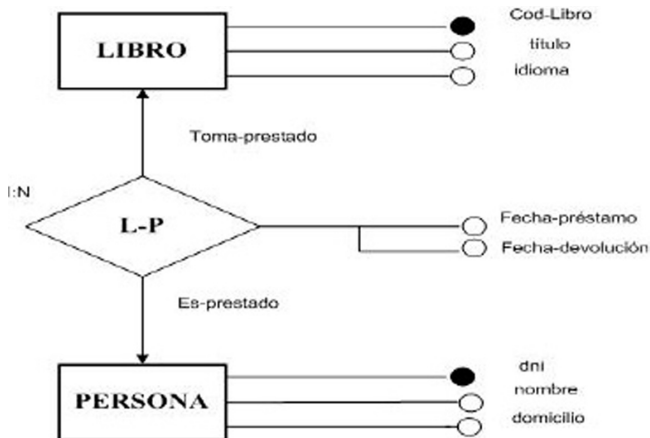
Atributo identificador principal:



Atributo identificador alternativo:



Los tipos de interrelación, al igual que los tipos de entidad pueden poseer atributos, así en la figura siguiente vemos que la interrelación A-C posee dos atributos Fecha-comienzo, Fecha-finalización y Calificación.



Restricciones en las interrelaciones

Existen cuatro tipos de restricciones en las interrelaciones:

- Restricciones de Exclusividad
- Restricciones de Exclusión
- Restricciones de Inclusividad
- Restricciones de Inclusión

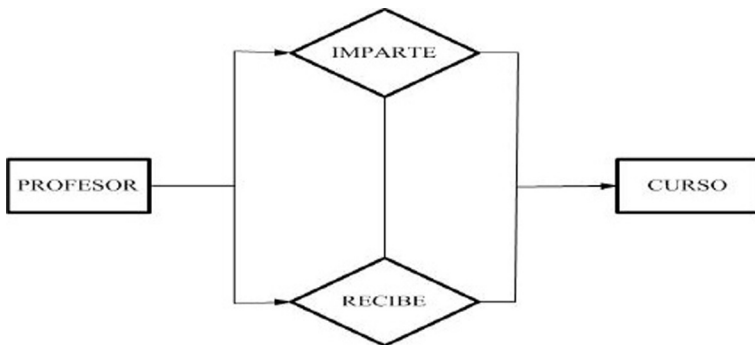
Restricción de Exclusividad

Dos o más interrelaciones son de exclusividad cuando cada ocurrencia de un tipo de entidad sólo puede combinarse con ocurrencias del otro tipo de entidad .

Los empleados, en función de sus capacidades, o son diseñadores de productos o los fabrican. No puede darse el caso de que un empleado sea diseñador y fabricante a la vez.

Restricción de Exclusión

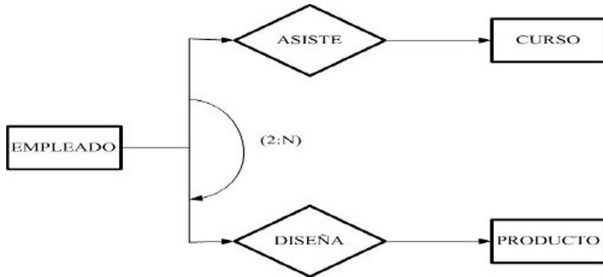
Se produce una restricción de exclusión cuando las ocurrencias de un tipo de entidad sólo pueden combinarse utilizando una interrelación. Veamos el siguiente ejemplo:



Un profesor no puede recibir e impartir el mismo curso, aunque al contrario que en la restricción anterior puede impartirlo o recibirlo.

Restricción de Inclusividad

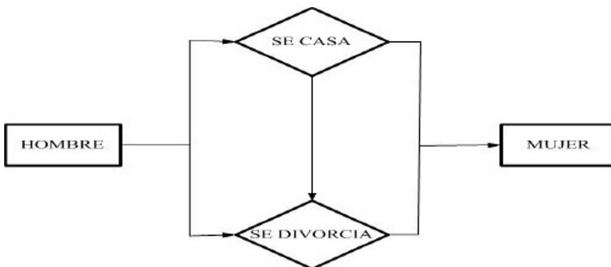
Se dice que una relación es de inclusividad cuando toda ocurrencia de un tipo de entidad que participa en una interrelación ha tenido que participar en la otra.



Para que un empleado pueda trabajar como diseñador de productos debe haber asistido, al menos, a dos cursos.

Restricción de Inclusión

Se establece una restricción de inclusión cuando toda ocurrencia de un tipo de entidad, para participar en la asociación con otro elemento de otra entidad mediante una interrelación, es necesario que ambos elementos estén asociados por una segunda interrelación. Por ejemplo:



Para que un hombre se divorcie de una mujer, previamente ha de haberse casado con ella.

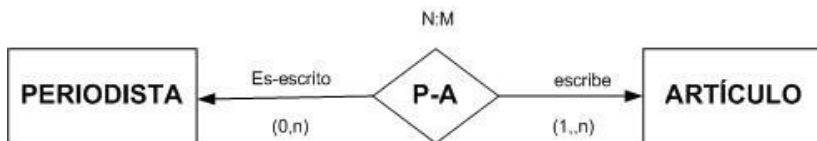
3.2.- Semántica de las interrelaciones

Cardinalidades

Las cardinalidades mínima y máxima de un tipo de interrelación son el número mínimo y máximo de ocurrencias de un tipo de entidad que pueden estar interrelacionadas con una ocurrencia del otro, u otros tipos de entidad que participan en el tipo de interrelación. Se representa gráficamente con la etiqueta colocada sobre el arco que une el tipo de entidad con el tipo de interrelación. Esta etiqueta será: (0,1), (1,1), (0,n) o (1,n) según corresponda.

La representación gráfica significa:
(cardinalidad mínima, cardinalidad máxima)

En la siguiente figura, la etiqueta (1,n) en ARTÍCULO significa que una ocurrencia de PERIODISTA puede estar relacionada con 1,2,...ó n ocurrencias de ARTÍCULO en el tipo de interrelación A-A. La etiqueta (0,n) en PERIODISTA significa que una ocurrencia de ARTÍCULO alcanza estar relacionada con 0,1,2,... ó n ocurrencias de PERIODISTA, es decir, un periodista escribe como mínimo un artículo(1) y como máximo muchos artículos (n). Un artículo puede ser escrito por ningún periodista (0), en el caso de que sea anónimo o por muchos (n).



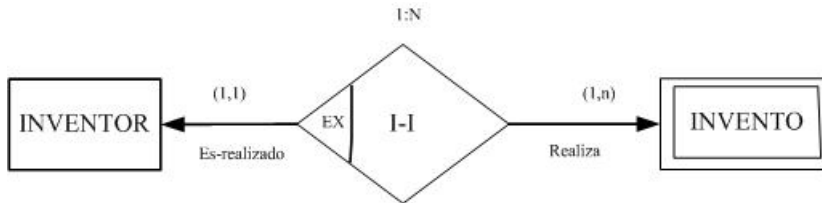
Vemos que el tipo de correspondencia, en este caso N:M coincide con la cardinalidad máxima de los tipos de entidades que asocian en la interrelación.

Dependencias

Los tipos de interrelación se clasifican, según los tipos de entidades que vinculen, y pueden ser de dos tipos:

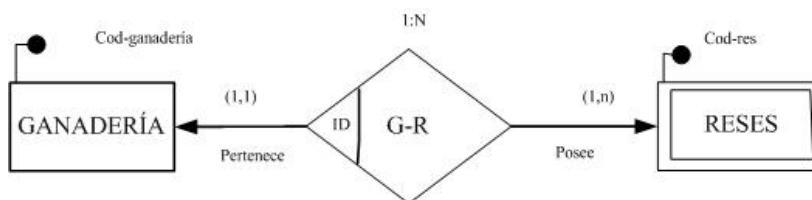
- **Regulares.**- cuando relaciona solamente tipos de entidades regulares.
- **Débiles.**- cuando relaciona un tipo de entidad débil con un tipo de entidad regular. Un tipo de interrelación débil siempre exige que las cardinalidades del tipo de entidad regular sean (1,1). Se representa añadiendo una “E” al rombo que representa la interrelación débil con debilidad por existencia. Dentro de este tipo de interrelación débil podemos distinguir:

- **Debilidad por existencia o Dependencia en existencia.**- cuando en la interrelación está relacionado un tipo de entidad regular con una débil, de forma que las ocurrencias del tipo de entidad débil no pueden existir sin la ocurrencia de la entidad regular de la que dependen. Si desaparece una ocurrencia de un tipo de entidad regular, todas las ocurrencias de la entidad débil que dependen de la existencia de la misma desaparecen con ella.. Se representa añadiendo la etiqueta “EX” al rombo que representa la interrelación. Veamos un ejemplo:



Vemos que el tipo de interrelación I-I asocia el tipo de entidad regular INVENTOR con el tipo de entidad débil INVENTO es una dependencia en existencia. No puede existir un invento, si no existe una persona que lo invente.

- **Debilidad por identificación o Dependencia en identificación.**- cuando además de dependencia en existencia, las ocurrencias del tipo de entidad débil no se pueden identificar sólo mediante sus propios atributos y precisan que se añada la clave de la ocurrencia de la entidad regular de la cual dependen. Se representa añadiendo la etiqueta “ID” al rombo que representa la interrelación. Veamos un ejemplo:



Vemos que el tipo de interrelación G-R, que asocia el tipo de entidad regular GANADERÍA con el tipo de entidad RESES, es dependiente en identificación, ya que un ejemplar determinado, además de depender en identificación, depende en existencia de una cierta ganadería, está identificado con la clave de la ganadería (Cod-Ganadería) del cual depende el ejemplar, más un código propio (Cod-res).

3.3.- Generalización y Herencia

La generalización es una abstracción que identifica una relación jerárquica que representa un tipo de entidad ES-UN subtipo de otro tipo de entidad representada a un nivel de abstracción mayor. Este tipo de relaciones entre tipos de entidad

implica la consideración de tipos de entidad (o supertipos) y de subtipos de entidad (clases, superclases y subclases de objetos).

Un subtipo de entidad es un tipo de entidad que mantiene un tipo de interrelación jerárquica con otro tipo de entidad, y que:

- Las propiedades y el comportamiento de los subtipos son heredados del tipo de entidad con el cual tienen un tipo de interrelación jerárquica. Para cada subtipo de entidad pueden ser redefinidas tanto las propiedades como el comportamiento del tipo de entidad con la que mantienen un tipo de interrelación jerárquica.
- Un tipo de entidad puede ser un subtipo para más de un tipo de entidad con las que puede mantener diferentes relaciones jerárquicas. A esto se denomina “herencia múltiple”, y permite que un tipo de entidad herede propiedades y comportamiento de más de otro tipo de entidad. La herencia múltiple puede provocar inconsistencias en las propiedades y/o comportamiento que se hereda, esto se debe solucionar mediante la redefinición de las herencias.

La especialización de un tipo de entidad en un conjunto de subtipos puede ser:

- **Especialización exclusiva.**- también se denomina “especialización sin solapamiento”, y representa el hecho de que una instancia del tipo de entidad más general sólo puede pertenecer o estar asociada a una y sólo una instancia de los subtipos de entidad especializados.
- **Especialización inclusiva.**- también se denomina “especialización con solapamiento”, y representa el hecho de que una instancia del tipo de entidad más general puede tener asociadas instancias de cualquiera de los subtipos.

También podemos hacer la siguiente clasificación:

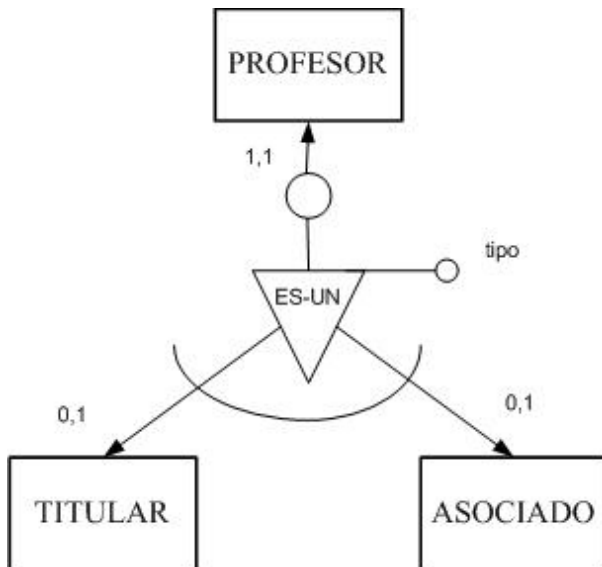
- **Especialización total.**- Las entidades reconocidas en el problema que se está representando son de alguno de los subtipos especializados, no existiendo entidades que no pertenezcan a alguno, varios o todos de estos subtipos de entidad.
- **Especialización parcial.**- Pueden existir entidades que pertenezcan al tipo de entidad y no a ninguno de los subtipos en los cuales este tipo de entidad está especializado.

Se pueden presentar cuatro tipos de interrelaciones jerárquicas:

- Total sin solapamiento
- Parcial sin solapamiento.
- Total con solapamiento.
- Parcial con solapamiento.

Veamos una serie de ejemplos:

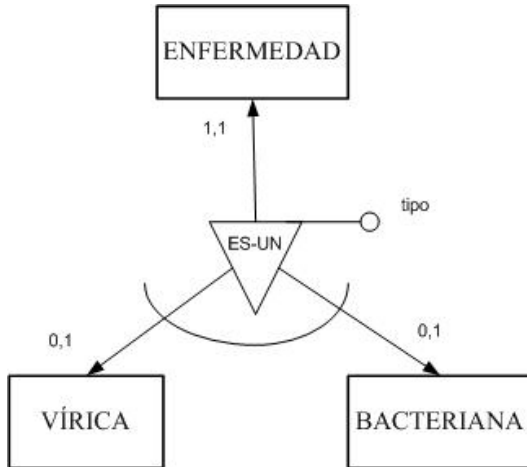
Ejemplo de Especificación Total y sin solapamiento:



Tenemos la entidad PROFESOR, la cual puede ser especializada en dos subtipos de entidad TITULAR y ASOCIADO. La entidad PROFESOR podrá pertenecer al subtipo TITULAR o al subtipo ASOCIADO necesariamente. No existirá una entidad PROFESOR que no sea de alguno de estos dos subtipos y además de forma exclusiva, por lo que una entidad pertenecerá a uno y sólo a uno de estos subtipos. Cada entidad de estos subtipos vendrá caracterizada por algún atributo o conjunto de atributos definidos para estos subtipos o heredados del tipo de entidad Persona, además del atributo *tipo*, el cual tiene la función de clasificador de las entidades PROFESOR en alguno de estos subtipos.

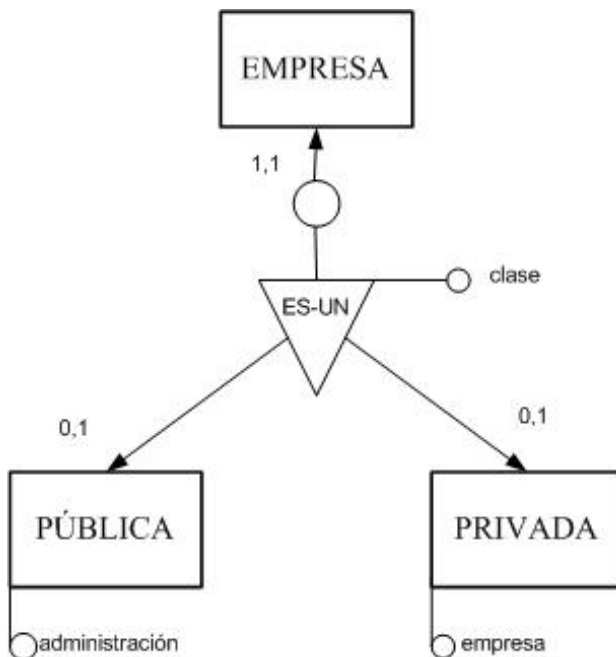
Ejemplo de Especificación Parcial y sin solapamiento:

Un tipo de entidad ENFERMEDAD puede ser especializada en dos subtipos VIRICA y BACTERIANA. Este diagrama permite representar un conjunto de entidades ENFERMEDAD las cuales pertenecen a alguno de los subtipos VÍRICA o BACTERIANA, pero que además aquellas que no puedan ser clasificadas en ninguno de estos subtipos, debido al desconocimiento del valor del atributo *tipo* utilizado como discriminador.



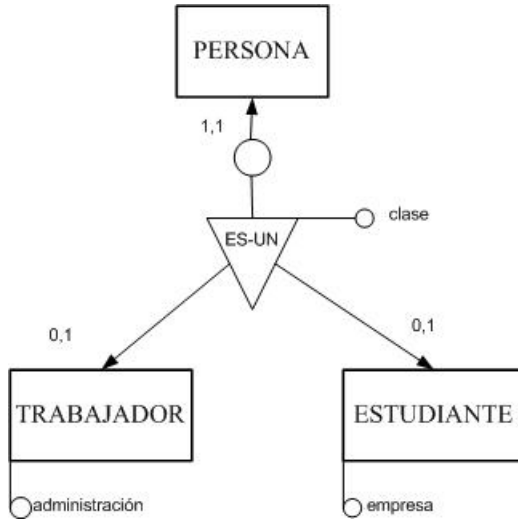
Ejemplo de Especificación Total y con solapamiento:

Un tipo de entidad EMPRESA se ha refinado en dos subtipos PÚBLICA y PRIVADA. Representa el hecho de que podrán existir entidades que pueden ser consideradas tanto pública como privada, o bien de ambos tipos al mismo tiempo, y también el hecho de que no podrán existir entidades que no puedan ser especializadas en alguno de estos dos subtipos.



Ejemplo de Especificación Parcial y con solapamiento:

El tipo de entidad PERSONA puede ser refinado en dos subtipos TRABAJADOR y ESTUDIANTE de forma parcial con solapamiento. Este ejemplo representa a una entidad PERSONA que puede ser del tipo TRABAJADOR y/o del tipo ESTUDIANTE y que además pueden existir entidades PERSONA que no puedan clasificarse en ninguno de estos dos subtipos.



Las cardinalidades en la generalización

La cardinalidad mínima y la cardinalidad máxima del tipo de entidad supertipo es siempre 1. Esto representa que una entidad supertipo puede especializarse en otros subtipos.

La cardinalidad máxima con la que participan los subtipos de entidad en el tipo de interrelación es 1. Esto está representando para cada entidad del supertipo una especialización de la misma.

Si la interrelación es sin solapamiento, los subtipos participan siempre con cardinalidad mínima 0, ya que una entidad del supertipo no puede pertenecer al mismo tiempo a más de un subtipo.

Si la interrelación es con solapamiento, los subtipos participan siempre con cardinalidad mínima 0 ó 1, ya que una entidad del supertipo puede a su vez pertenecer a cualquiera de los subtipos.

3.4.- Dinámica del Modelo Entidad-Interrelación

Chen no se ocupó de la parte dinámica del modelo ME-R. Posteriormente en varios trabajos se han propuesto lenguajes basados en el ME-R. Estos lenguajes, permiten definir la parte estática del modelo (entidades, interrelaciones, etc.) y formular consultas a la base de datos con sentencias que se parecen al lenguaje natural y son sencillas de realizar.

4.- EL MODELO RELACIONAL

A finales de los años 60, Edgar Frank Codd definió las bases del Modelo Relacional, donde los datos se estructuran lógicamente en forma de relaciones (tablas). Codd, que era un experto matemático, utilizó una terminología perteneciente a las matemáticas, en concreto de la teoría de conjuntos y de la lógica de predicados. El objetivo fundamental del modelo relacional es mantener la independencia de esta estructura lógica respecto al modo de almacenamiento y a otras características de tipo físico.

La estructura básica del modelo relacional es la **relación o tabla**. Todos los datos de la base de datos se representan en forma de tabla o relación cuyo contenido varía con el tiempo. Una relación o tabla se representa gráficamente como una estructura rectangular, formada por filas o columnas. Cada columna almacena información sobre una propiedad determinada de la tabla o relación. A estas columnas también se las denomina **atributos**. Los atributos pueden aparecer en la relación en cualquier orden.

Cada fila de la tabla se denomina **tupla**, y representa una ocurrencia de la relación. Cada tupla almacena los valores que toma cada uno de los atributos, para cada ocurrencia de la relación.

Cada relación o tabla está formada por dos partes:

- **Cabecera**.- conjunto fijo de pares *atributo:dominio*, donde cada atributo pertenece a un único dominio subyacente y todos los atributos son distintos, es decir, no hay dos atributos que se llamen igual.

- **Cuerpo.**- es un conjunto variable de tuplas. Cada tupla es un conjunto de pares *atributo:valor*, cuyo valor varía con el tiempo.

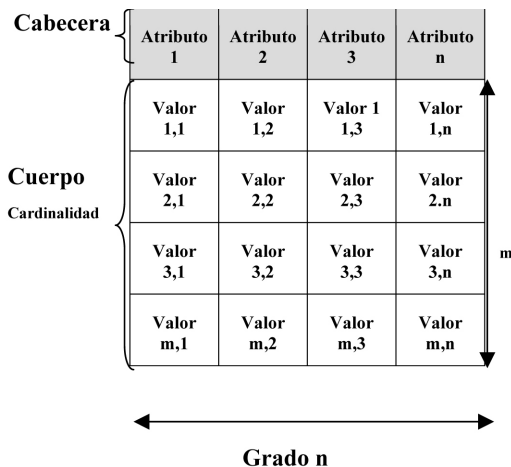
Otros términos importantes en la relación son:

- **Cardinalidad.**- Es el número de filas o tuplas de la relación.
- **Grado.**- Es el número de atributos o columnas de la relación.

4.1.- Propiedades de las relaciones

Las relaciones tienen las siguientes características:

- Cada relación tiene un nombre y éste es distinto del nombre de todas las demás.
- Los valores de los atributos son atómicos: en cada tupla, cada atributo toma un solo valor. Se dice que las relaciones están *normalizadas*.
- No hay dos atributos que se llamen igual.
- El orden de los atributos no importa: los atributos no están ordenados.
- Cada tupla es distinta de las demás: no hay tuplas duplicadas.
- El orden de las tuplas no importa: las tuplas no están ordenadas.



Ejemplo:

EMPLEADO

DNI	NOMBRE	TELEFONO
20610777	SÁNCHEZ LÓPEZ, JAVIER	957-45-77-00
66884545	MORENO FUENTES, ANGEL	91-347-99-88
34233325	CARDENAS DIEZ, MÓNICA	954-33-44-22
35678102	DIAZ ESPINOSA, MARIA	952-99-55-66

Con respecto a la dinámica del modelo, se propone un conjunto de operadores que se aplican a las relaciones. Algunos de estos operadores son clásicos de la teoría de conjuntos, no hay que olvidar que una relación se define matemáticamente como un conjunto, mientras que otros operadores fueron introducidos específicamente para el Modelo Relacional. Todos estos operadores constituyen el **Álgebra Relacional**.

Un **dominio** D es un conjunto finito de valores homogéneos y atómicos V_1, V_2, \dots, V_n caracterizado por un nombre. Se dice que los valores son **homogéneos** porque son todos del mismo tipo, y **atómicos** porque son indivisibles.

Todo dominio ha de tener un nombre, por el cual nos podemos referir a él, y un tipo de datos. Los dominios pueden definirse por **intención** o por **extensión**. Veamos unos ejemplos:

Dominio de las edades de las personas activas:

Intención: $D(V_1, V_2, \dots, V_i, \dots, V_n) / \forall i \ 18 \leq V_i \leq 65$

Dominio de los cargos de los empleados:

Extensión: $D(\text{"Director General"}, \text{"Director de Operaciones"}, \text{"Comercial"}, \text{"Jefe de Personal"}, \text{"Jefe de Proyecto"}, \text{"Analista"}, \text{"Programador"}, \text{"Secretaria"}).$

Cada atributo de una relación toma sus valores de un único dominio, denominado "**dominio subyacente**". Varios atributos pueden tener el mismo dominio subyacente.

Un **dominio compuesto** se puede definir como una combinación de dominios simples a los que se les puede aplicar ciertas restricciones de integridad.

Por ejemplo:

Dominio Fecha (Dominio Día, Dominio Mes, Dominio Año).

Igual que podemos definir dominios compuestos, también podemos definir **atributos compuestos**. Por ejemplo, el atributo *Fecha-Nacimiento* tomaría sus valores del dominio *Fecha*. Los atributos compuestos pueden ser tratados como valores atómicos si lo necesita el usuario.

En la tabla que hay a continuación vemos la equivalencia que existe entre los términos formales utilizados en el modelo relacional y los informales:

TERMINO FORMAL	TERMINO INFORMAL
Relación	Tabla
Atributo	Columna o campo
Grado	Nº de Columnas
Tupla	Fila o registro
Cardinalidad	Nº de filas
Clave Primaria	Identificador único
Dominio	Conjunto de valores legales

4.2.- Definición formal de relación

Nombre: Permite identificar la relación.

Cabecera de la relación: Conjunto de n pares atributo-dominio subyacente $\{(A_i:Di)\}^n$, donde n es el grado. El conjunto A de atributos sobre los que se define la relación se llama **contexto** de la misma. La cabecera es invariante, es decir, no varía con el tiempo.

Cuerpo de la relación: Conjunto de m tuplas $\{t_1, t_2, \dots, t_m\}$, donde cada tupla es un conjunto de n pares atributo-valor $\{(A_i:V_{ij})\}$, siendo V_{ij} el valor j del dominio Di asociado al atributo A_i ; el número de tuplas m es la cardinalidad. El cuerpo varía con el tiempo, al igual que la cardinalidad.

El esquema de la relación: Se denota por $R(\{A_i:Di\}_n)$, donde R es el nombre de la relación.

Donde A_i es el atributo y Di es el dominio subyacente. El esquema de la relación representa la parte estática y defini-

toria, y se denomina también *intención*. Se corresponde con el tipo de entidad en el ME-R.

Estado de relación r®: Se denomina simplemente relación y está constituido por el esquema y el cuerpo de la relación, siendo el cuerpo el conjunto de tuplas que, en un instante dado, satisface el correspondiente esquema de la relación.

Veamos unos ejemplos:

Esquema de la relación (Intención):

ALUMNO(Nombre: *Nombre*, Nacionalidad: *Nacionalidades*, Teléfono: *Teléfonos*).

Relación (Extensión, Estado u Ocurrencia):

ALUMNO

Nombre	Nacionalidad	Teléfono
Sánchez Pérez, Ángel	Española	91-883-66-44-22
López Sanz, Mario	Española	91-444-33-22-11
Caballero Delgado, Manuel	Mexicana	953-77-66-55
Moreno Berlanga, Gabriel	Ecuatoriano	957-33-22-11

Clases de Relación

Las relaciones pueden clasificarse en:

- **Persistentes.**- Aquellas cuya definición (esquema de relación) permanece en la base de datos, borrándose solamente mediante una acción explícita del usuario. Las relaciones se dividen en:
 - Relaciones base.- Existen por sí mismas, no en función de otras relaciones, y se crean especificando explícitamente su esquema de relación (nombre y conjunto de pares: atributo-dominio). Sus extensiones (ocurrencias de la relación), al igual que su definición, también se encuentran almacenadas.
 - Vistas.- Son relaciones derivadas que se definen dando un nombre a una expresión de consulta. Son relaciones virtuales, ya que no tienen datos almacenados y sólo contienen la definición en términos de otras relaciones.
- **Temporales.**- Una relación temporal desaparece de la base de datos sin necesidad de una acción de borrado específica del usuario.

Claves

- **Clave candidata.**- Es un conjunto finito, no vacío de atributos que identifican unívoca y mínimamente cada tupla de la relación. Por definición una relación tiene siempre, al menos, una clave candidata. Una relación puede tener más de una clave candidata, y entre ellas podemos distinguir:
 - **Clave primaria.**- Es la clave candidata que el usuario elige, por consideraciones ajenas al modelo, para identificar las tuplas de la relación.
 - **Clave alternativa.**- Son las claves candidatas que no han sido elegidas como clave primaria.
- **Clave externa.**- de una relación R_2 , es un conjunto no vacío de atributos cuyos valores han de coincidir con los valores de la clave candidata de una relación R_1 . La clave externa y la correspondiente clave primaria han de estar definidas sobre el mismo dominio.

4.3.- Restricciones

Las restricciones son estructuras u ocurrencias no permitidas. Los datos de la base de datos, han de cumplir las restricciones, para poder ser una ocurrencia válida del esquema. Hay dos tipos de restricciones:

- **Restricciones inherentes al modelo relacional**

Son las restricciones que impone el propio modelo, no son definidas por el usuario y son las siguientes:

- No hay dos tuplas iguales, esto implica, la obligatoriedad de la clave primaria.
- El orden de las tuplas no es significativo.
- El orden de los atributos no es significativo.
- Cada atributo sólo puede tomar un único valor del dominio sobre el que está definido, no admitiéndose por tanto los grupos repetitivos.

- **Restricciones semánticas**

También se llaman restricciones de usuario, y son facilidades que el modelo ofrece a los usuarios para que éstos puedan reflejar en el esquema, lo más fielmente posible, la semántica del mundo real. Las principales restricciones semánticas del modelo relacional son:

- **Clave primaria.**- Permite declarar un atributo o conjunto de atributos como clave primaria de la relación. Sus valores no se podrán repetir, ni ser nulos (integridad de clave). La obligatoriedad de la clave primaria es una restricción inherente al modelo relacional; pero la declaración de un atributo como clave primaria de una relación es una restricción semántica.
- **Unicidad.**- Indica que los valores de un conjunto de atributos no pueden repetirse en una relación. Permite definir claves alternativas.

- **Obligatoriedad.**- Indica que el conjunto de atributos no admite valores nulos.
- **Integridad referencial.**- Si una relación R2 (relación que referencia) tiene un atributo que es una clave candidata de la relación R1 (relación referenciada), todo valor de dicho atributo debe concordar con un valor de la clave candidata referenciada de R1 o bien ser nulo. El atributo es clave externa de la relación R2. La clave externa puede ser también parte o total de la clave primaria de R2.

También hay que determinar las consecuencias que pueden tener ciertas operaciones, como el borrado y la modificación, realizadas sobre tuplas de la relación referenciada, y son las siguientes:

- **Operación restringida (restrict).**- El borrado o modificación de tuplas que contiene la clave referenciada sólo se permite si no existen tuplas con este valor en la relación que contiene la clave externa.
- **Operación con transmisión en cascada (cascade).**- El borrado o modificación de tuplas que contiene la clave referenciada, lleva consigo el borrado o modificación en cascada de las tuplas de la relación que contiene la clave externa.
- **Operación con puesta a nulos (set null).**- El borrado o modificación de tuplas que contiene la clave referenciada lleva consigo poner a nulos los valores de las claves externas de la relación que referencia.
- **Operación con puesta a valor por defecto (set default).**- El borrado o modificación de tuplas que contiene la clave referenciada lleva consigo poner el valor por

defecto a la clave externa de la relación que referencia, valor por defecto que habría sido definido al crear la tabla.

Además de estas restricciones existen las llamadas *restricciones de rechazo*, en las que el usuario formula una condición mediante un predicado definido sobre un conjunto de atributos, de tuplas o dominios, el cual debe ser verificado por los correspondientes objetos en toda operación de actualización para que el nuevo estado constituya una ocurrencia válida del esquema.

Existen dos restricciones del rechazo en el modelo relacional, según que la condición afecte a uno o a varios elementos de la base de datos:

- **Verificación.**- En toda operación de actualización, comprueba si el predicado es verdadero o falso y, en el segundo caso, rechaza la operación. La restricción de verificación se define sobre un único elemento y puede o no tener nombre.
- **Aserción.**- Igual que la anterior, pero afecta a varios elementos (por ejemplo dos relaciones distintas) y a su definición. Siempre ha de tener nombre, ya que la aserción es un elemento más del esquema.

4.4.- Los valores nulos

Definición de valor nulo

El valor nulo se puede definir como una marca utilizada para representar información desconocida, inexistente, no proporcionada, inaplicable, indefinida, no válida, etc.

Los valores nulos son necesarios cuando se presentan los siguientes casos:

- Cuando se desean crear tuplas con algunos atributos desconocidos en ese momento.

- Cuando se desea añadir un nuevo atributo a una relación existente, pero que en el momento de añadirse no tiene ningún valor para las tuplas de esa relación.
- Cuando hay atributos inaplicables a ciertas tuplas.

Tratamiento de los valores nulos

El tratamiento de los valores nulos es complejo. Se puede definir una lógica trivaluada para contemplar la existencia de valores nulos a la hora de determinar el resultado de las operaciones lógicas: AND, OR y NOT.

X	Y	X AND Y	X OR Y	NOT X
V	V	V	V	F
V	F	F	V	-
V	N	N	V	-
F	V	F	V	V
N	V	N	V	N
F	N	F	N	-
N	F	F	N	-
F	F	F	F	-
N	N	N	N	-

Se ha introducido un nuevo tipo de operador denominado MAYBE (quizás) y que corresponde a la pregunta ¿Es nulo?, siendo su tabla de verdad la siguiente:

MAYBE	
V	F
F	F
N	V

Cuando se trabaja con valores nulos en algún operando es preciso definir:

- **Operaciones aritméticas.**- Cuando en una suma, resta, multiplicación o división alguno de los operandos es nulo, el resultado es también nulo.
- **Operaciones de comparación.**- Operadores:
ES_NULO: es cierto cuando el operando es nulo, y falso en caso contrario
SI_NULO: que se aplica a dos operandos y devuelve el valor del primero, excepto si éste es nulo, en cuyo caso devuelve el valor del segundo.

Los valores nulos en claves han de cumplir lo siguiente:

- La clave candidata no puede presentar valores nulos.
- Un atributo utilizado para el cómputo de un índice sí puede presentar valores nulos aunque se haya definido como único.
- Una clave ajena puede contener valores nulos.

4.5.- Las doce reglas de Codd

Codd estableció doce reglas, que se exponen a continuación:

1. **La regla de la información.**- Toda la información de una base de datos relacional está representada explícitamente a nivel lógico y exactamente de un modo: mediante valores en tablas.

2. **Regla de acceso garantizado.**- Todos y cada uno de los datos (valor atómico) de una base de datos relacional se garantiza que sean lógicamente accesibles recurriendo a una combinación de nombre de tabla, valor de clave primaria y nombre de columna.
3. **Tratamiento sistemático de valores nulos.**- Los valores nulos han de ser tratados sistemáticamente por el sistema, el cual ha de ofrecer las facilidades necesarias para su tratamiento, el cual ha de ofrecer la facilidades necesarias para su tratamiento.
4. **Catálogo activo en línea basado en el modelo relacional.**- La representación de la metainformación (descripción de la base de datos), debe ser igual a la de los otros datos, y su acceso de pode realizarse por medio del mismo lenguaje relacional que se utiliza para los demás datos.
5. **Sublenguaje completo de datos.**- Debe existir un lenguaje que permite un completo manejo de la base de datos (definición de datos, definición de vistas, manipulación de datos, restricciones de integridad, autorizaciones y gestión de transacciones).
6. **Actualización de vistas.**- Todas las vistas que sean teóricamente actualizables, son también actualizables por el sistema.
7. **Inserción, actualización y supresión de alto nivel.**- Todas las operaciones de manipulación de datos debe operar sobre conjunto de filas.
8. **Independencia física de los datos.**- El acceso lógico a los datos debe mantenerse incluso cuando cambien los métodos de acceso o la forma de almacenamiento.

9. **Independencia lógica de los datos.**- Los programas de aplicación no se verán afectados por los cambios permitidos que se realicen en las tablas.
10. **Independencia de la integridad.**- Las restricciones de integridad específicas para una base de datos relacional particular deben ser definibles en el sublenguaje de datos relacional y almacenables en el catálogo, no en los programas de aplicación.
11. **Independencia de la distribución.**- Debe existir un sublenguaje de datos que pueda soportar las bases de datos distribuidas sin alterar los programas de aplicación cuando se distribuyan los datos por primera vez o se redistribuyan éstos posteriormente.
12. **Regla de no subversión.**- Si un sistema relacional tiene un lenguaje de bajo nivel (un solo registro cada vez), dicho lenguaje no puede ser utilizado para subvertir o suprimir las reglas de integridad y las restricciones expresadas en el lenguaje relacional de nivel superior (múltiples registros a la vez).

5.- CÓMO PASAR DEL MODELO ENTIDAD-INTERRELACIÓN AL MODELO RELACIONAL

La traducción del Modelo Entidad-Interrelación en un Modelo Relacional consiste en convertir el Esquema Conceptual en un Esquema Lógico Relacional aplicando una serie de reglas de transformación.

5.1.- Pasos iniciales de la traducción

Debemos preparar los esquemas conceptuales y para ello utilizaremos unas reglas denominadas **Reglas Preparatorias**. Estas consisten en la aplicación de la primera forma normal a los objetos del Esquema Conceptual. Vamos a eliminar las siguientes anomalías:

3. Los atributos correspondientes a los tipos de entidad e interrelación que presenten valores múltiples.
4. Los atributos correspondientes a los tipos de entidad e interrelación que sean compuestos.

Reglas:

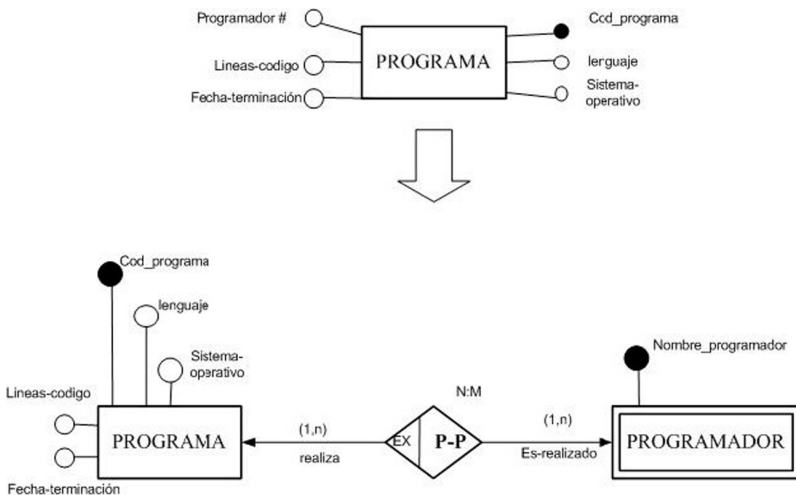
Regla de Eliminación de atributos múltiples (#):

Los atributos que pueden tomar más de un valor en el dominio en el cual están definidos, se transformarán en un tipo de entidad débil por existencia, que mantendrá una relación uno a muchos (1:N) con el tipo de entidad sobre el cual estaba definido o los tipos de entidad que mantenían un tipo de interrelación si el atributo múltiple estaba definido sobre el tipo de interrelación. Este tipo de entidad débil, tendrá como

propiedades el atributo múltiple por el cual la regla se ha aplicado. Si el atributo del tipo de entidad débil creado no pudiera identificar sin ambigüedad a las entidades de este tipo, entonces se procederá de alguna de las dos formas siguientes:

1. El tipo de entidad débil creado se considera que es débil por identificación con respecto al tipo de entidad con el que mantiene relación, heredando sus atributos identificadores. Si el atributo múltiple estaba definido sobre un tipo de interrelación, heredará los identificadores de los tipos de entidad que participaban en este tipo de interrelación.
2. Se añadirá un nuevo atributo, que permite identificar sin ambigüedad a las entidades de este tipo de interrelación.

Veamos un ejemplo:

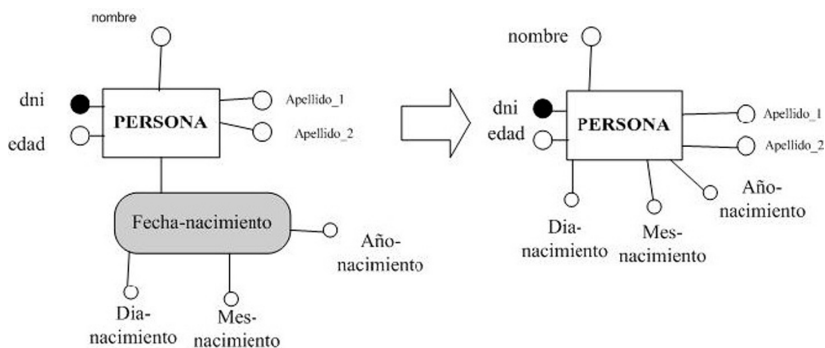


El tipo de entidad PROGRAMA tiene un atributo múltiple “programador”. Al aplicar esta regla, el atributo *programador* se convierte en un tipo de *entidad débil con dependencia en existencia*. El tipo de entidad PROGRAMADOR tiene como único atributo *Nombre_programador*.

Regla de Eliminación de atributos compuestos:

Todos los atributos compuestos pertenecientes a los tipos de entidad o a los tipos de interrelación deben ser descompuestos en los atributos simples que formen parte o intervengan en los atributos compuestos correspondientes. Los atributos simples quedan definidos en el mismo dominio e interviniendo de la misma forma en el tipo de entidad o interrelación.

Veamos un ejemplo:



5.2.- Transformación de los tipos de entidad

Todos los tipos de entidad del esquema conceptual se transformarán en tablas o relaciones en el esquema relacional. Seguirán teniendo el mismo número y tipo de atributos, así como las características de identificador de estos atributos.

Veamos un ejemplo:

Partiendo del último ejemplo quedaría así

PERSONA (DNI, edad, nombre, apellido_1, apellido_2, Día_nacimiento, Mes_nacimiento, Año_nacimiento).

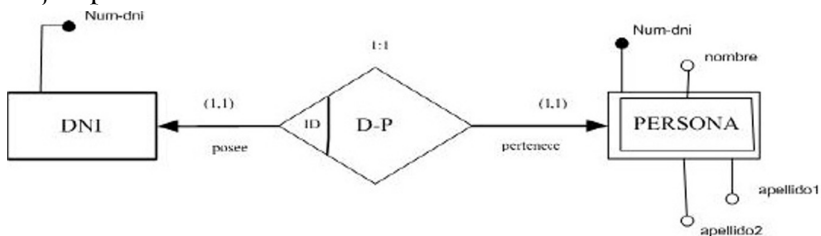
Los atributos que aparecen subrayados indican que son la clave principal.

5.3.- Transformación de los tipos de interrelación 1:1

Según la forma en que participa cada tipo de entidad en el tipo de interrelación será necesario aplicar diferentes reglas:

- Si ambos tipos de entidad participan en el tipo de interrelación con las cardinalidades (1,1) entonces:
 - Si los dos tipos de entidad tienen el mismo identificador:
 - a) Los dos tipos de entidad se transforman en una única tabla formada por la agregación de los atributos de los dos tipos de entidad.
 - b) La clave de la tabla es el identificador de los tipos de entidad.

Ejemplo:



Obtendríamos las relaciones siguientes:

DNI(Num-dni)

PERSONA(**Num-dni**, nombre, apellido1,apellido2)

Otra posible solución es:

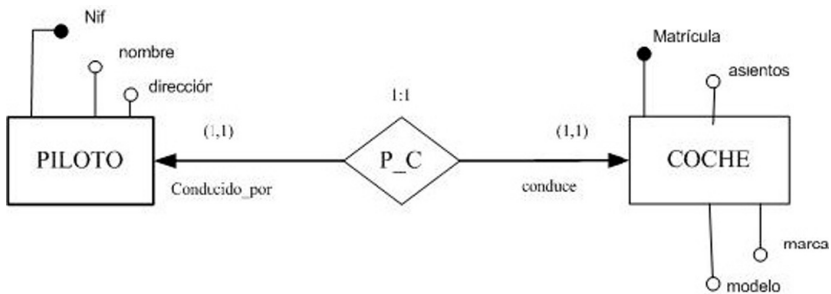
DNI(Num-dni, nombre, apellido1,apellido2)

· Si los tipos de entidad tienen diferente identificador, cada tipo de entidad se transforma en una tabla y,

1. Cada tabla tendrá como clave principal el identificador de cada uno de los tipos de entidad de los cuales se deriva.

2. Se produce el **fenómeno de Propagación de Clave**, donde cada tabla tendrá como clave externa o ajena al identificador del otro tipo de entidad con el cual está relacionado. La clave externa o ajena se representa en este libro en letra **negrita**.

Ejemplo:



Las relaciones que se obtienen son:

PILOTO(Nif, nombre, dirección, **matrícula**)

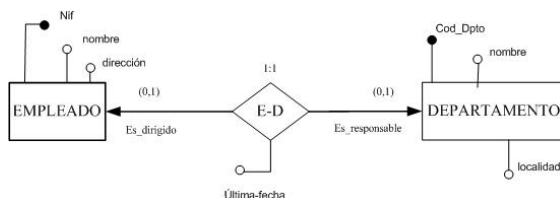
Aquí el atributo **matrícula** es clave ajena o externa.

COCHE (Matrícula, asientos, marca, modelo, **Nif**)

Aquí el atributo **Nif** es clave ajena o externa.

- Si los dos tipos de entidad tienen la misma clave, pero uno de ellos es un tipo de entidad débil, entonces se procede de alguna de las dos formas expuestas anteriormente en función de los requerimientos funcionales.
- Si la interrelación es binaria y alguno de los tipos de entidad participan de forma parcial, entonces, cada tipo de entidad se transforma en una tabla por aplicación de la Regla de Eliminación de atributos múltiples, y se procede de alguna de las formas siguientes:
 1. Se crea una tabla por cada uno de los tipos de entidad y se produce el **fenómeno de Propagación de Clave**, donde el identificador del tipo de entidad que participa totalmente, pasa como atributo (clave ajena o externa) de la tabla correspondiente a la transformación del otro tipo de entidad. Este atributo podrá tomar valores nulos para diferentes tuplas de esta tabla, no generándose ninguna tabla para el tipo de interrelación.
 2. Se crea una tabla por cada uno de los tipos de entidad y además se crea una nueva tabla correspondiente al tipo de interrelación formada por los atributos identificadores de los dos tipos de entidad. La clave será el identificador del tipo de entidad que participa de forma parcial en la interrelación. Los atributos de esta tabla mantendrán referencias con las claves de las tablas correspondientes a la transformación de los tipos de entidad.

Ejemplo:



Si aplicamos el punto 1 quedaría el esquema relacional como sigue:

EMPLEADO(Nif, nombre, dirección)

DEPARTAMENTO(Cod-Dpto, nombre, localidad, Nif)

Si aplicamos el punto 2 quedaría el esquema relacional como sigue:

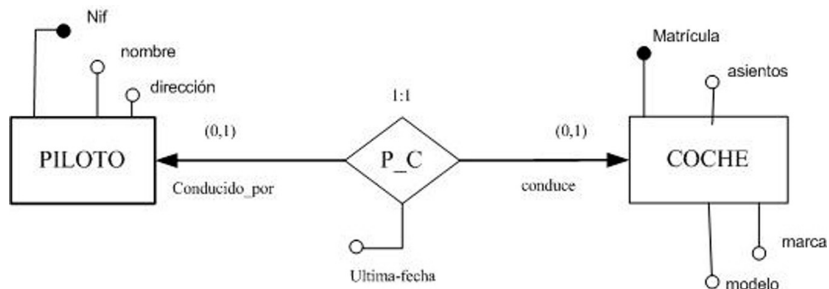
EMPLEADO(Nif, nombre, dirección)

DEPARTAMENTO(Cod-Dpto, nombre, localidad)

EMP-DPTO(Nif, Cod-Dpto)

· Si ambos tipos de entidad participan de forma parcial, cada uno de ellos se transformará en una tabla y se procede de alguna de las formas siguientes:

1. Se produce el **fenómeno de Propagación de Clave**, donde el identificador de cada tipo de entidad, pasa como atributo (clave ajena o externa) de la tabla correspondiente a la transformación del otro tipo de entidad.
2. Se crea una nueva tabla correspondiente al tipo de interrelación y cuyos atributos serán los identificadores de los dos tipos de entidad. La tabla tendrá como clave principal el identificador de uno de los tipos de entidad y, necesariamente, se definirá como clave alterna al identificador del otro tipo de entidad.



El esquema relacional resultante para el caso 1 sería:

PILOTO(Nif, nombre, dirección, **Matrícula**)

COCHE(Matrícula, asientos, marca, modelo, **Nif**)

El esquema relacional resultante para el caso 1 sería:

PILOTO(Nif, nombre, dirección)

COCHE(Matrícula, asientos, marca, modelo)

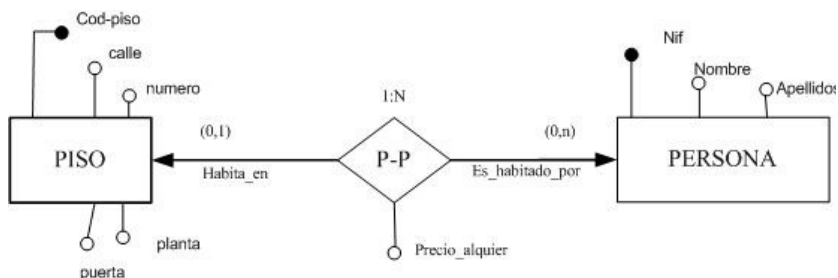
P-C(**Nif**, **matrícula**)

5.4.- Transformación de los tipos de interrelación

1:N

Según las cardinalidades mínimas con las cuales participa cada tipo de entidad tendremos:

- Si ambos tipos de entidad participan de forma total, o el tipo de entidad con cardinalidad máxima muchos participa de forma parcial, entonces cada tipo de entidad se transforma en una tabla. También se produce el fenómeno de Propagación de Clave, de forma que el identificador del tipo de entidad que tiene como cardinalidad máxima n será clave ajena en la tabla correspondiente al otro tipo de entidad. Si el tipo de interrelación tuviera atributos, éstos pasarían a formar parte de la tabla correspondiente al tipo de entidad que participa con cardinalidad máxima n.



El esquema relacional quedaría:

PROFESOR(Nif, nombre, dirección, **Cod-Dpto**, función)

DEPARTAMENTO(Cod-Dpto, Nombre-Dpto,)

- Si el tipo de entidad que tiene cardinalidad máxima uno participa de forma parcial, es decir, tiene cardinalidad (0,1), cada tipo de entidad se transforma en una tabla por la regla de Transformación de los Tipos de Entidad. Se genera además una nueva tabla que corresponde al tipo de interrelación, y tendrá los identificadores de los tipos de entidad que intervienen en el tipo de interrelación y todos los atributos asociados al tipo de interrelación. La clave principal de esta tabla será el identificador del tipo de entidad que participa con cardinalidad máxima muchos. Serán claves ajenas o externas los atributos identificadores de los dos tipos de entidad.

Ejemplo:

El esquema relacional que se deriva es:

PERSONA(Nif, Nombre, Apellidos)

PISO(Cod-piso, calle, número, planta, puerta)

P-P(Nif , Cod-piso , Precio_alquiler)

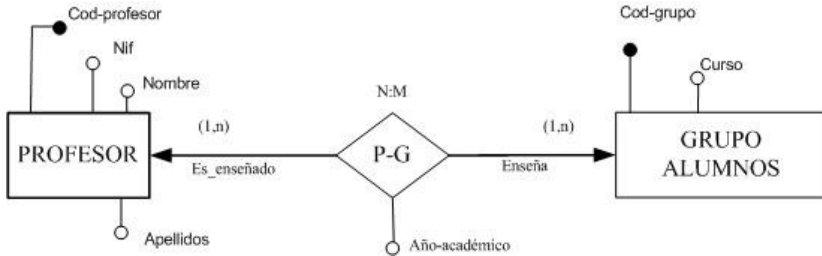
5.5.- Transformación de los tipos de interrelación

N:M

Siempre se aplica la siguiente regla:

Cada tipo de entidad se transforma en una tabla por la regla de Transformación de los Tipos de Entidad. Además se crea una nueva tabla correspondiente al tipo de interrelación. Esta tabla tendrá los identificadores de los tipos de entidad que intervienen en la interrelación y por todos los atributos asociados al tipo de interrelación. La clave principal de esta tabla será la agregación de los atributos identificadores correspondientes a los tipos de entidad que intervienen en el tipo de interrelación.

Ejemplo:



El esquema relacional que se obtendría sería:

PROFESOR(Cod-profesor, Nif, Nombre, Apellidos)

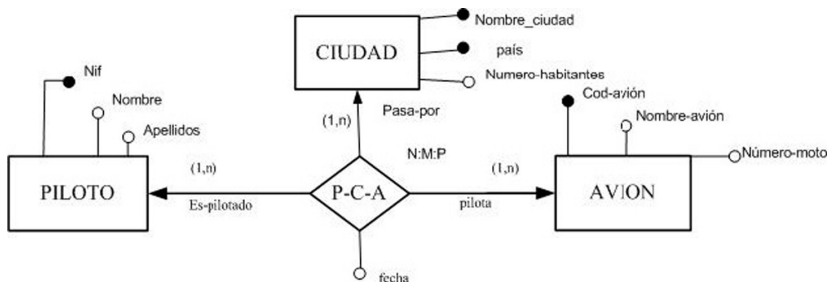
GRUPOALUMNOS(Cod-grupo, Curso)

P-G(**Cod-profesor**, **Cod-grup**, Año-académico)

5.6.-Transformación de los tipos de interrelaciones n-arias

Cuando en un tipo de interrelación intervienen más de dos tipos de entidad se debe aplicar la regla anterior(Regla de Transformación de los tipos de interrelación muchos a muchos N:M)

Ejemplo:



El esquema relacional quedará:

PILOTO(Nif, Nombre, Apellidos)

CIUDAD(Nombre-ciudad, país, número-habitantes)

AVIÓN(Cod-avión, nombre-avión, número-motores)

P-C-A (Nif, Nombre-ciudad, país, Cod-avión, fecha)

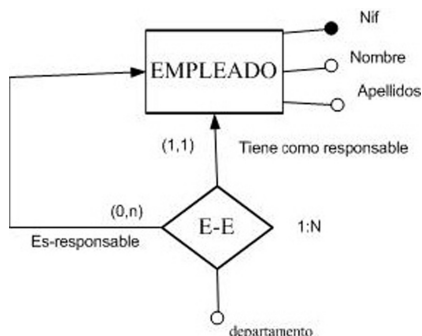
Esta regla que acabamos de aplicar debe utilizarse sólo cuando todos los tipos de entidad participan en el tipo de interrelación con cardinalidad máxima muchos. Si algún tipo de entidad participa con cardinalidad máxima uno, el identificador de este tipo de entidad no formará parte de la clave principal de la tabla correspondiente al tipo de interrelación.

5.7.-Transformación de los tipos de interrelaciones reflexivas

Un tipo de interrelación reflexiva o recursiva es aquella que se produce entre un mismo tipo de entidad consigo misma. El mismo tipo de entidad desempeña dos papeles en el tipo de interrelación. Pueden presentarse dos casos:

1. El tipo de entidad participa en uno de sus papeles con cardinalidad máxima uno, y en el otro papel con cardinalidad máxima muchos, o bien ambos participan con cardinalidad máxima uno.

Ejemplo:



Se procederá de una de las dos formas siguientes:

1.1.- Se crea una tabla para el tipo de entidad, y en esta tabla se añade como clave externa (fenómeno de propagación de clave) el identificador del tipo de entidad correspondiente al empleado que tiene como responsable otros. Se añade además el atributo del tipo de entidad del empleado que es responsable de él. Si el tipo de interrelación tiene atributos propios también se añaden (departamento).

El esquema quedaría:

EMPLEADO(Nif_tieneresponsible, Nombre, Apellidos, Nif_esresponsable, departamento)

1.2.- Se crea una tabla para el tipo de entidad y otra tabla para el tipo de interrelación. Esta última tabla tiene como clave principal el identificador del tipo de entidad que representa el empleado que es responsable de otros empleados. Además esta tabla estará formada por los atributos del tipo de interrelación (departamento) y se añadirá el identificador del tipo de entidad que representa al empleado que tiene como responsable otro empleado.

EMPLEADO(Nif, Nombre, Apellidos)

E-E(Nif-esresponsable, departamento, **Nif-tieneresponsible**)

2.- El tipo de entidad participa con cardinalidad máxima muchos en sus dos papeles.

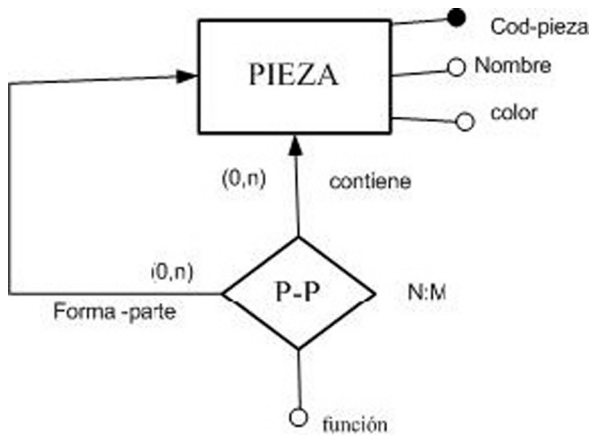
En el caso segundo, se aplica la regla de Transformación de los tipos de interrelación de muchos a muchos (N:M). Así se crea una tabla para el tipo de entidad y otra tabla para el tipo de interrelación, siendo los atributos los siguientes:

- El identificador del tipo de entidad que desempeña uno de los papeles en el tipo de interrelación.

- El identificador del tipo de entidad que desempeña el otro papel en el tipo de interrelación.
- Los atributos del tipo de interrelación.

La clave principal de esta tabla será el identificador del tipo de entidad por duplicado.

Ejemplo:



El esquema relacional quedaría:

PIEZA(Cod-pieza, Nombre, color)

P-P(Cod-pieza-continente, Cod-pieza-contenido, función)

5.8.- Eliminación de las Relaciones Jerárquicas

Para la eliminación de las relaciones jerárquicas podemos aplicar tres reglas diferentes:

- Eliminación del supertipo.
- Eliminación de los subtipos.
- Eliminación de la jerarquía.

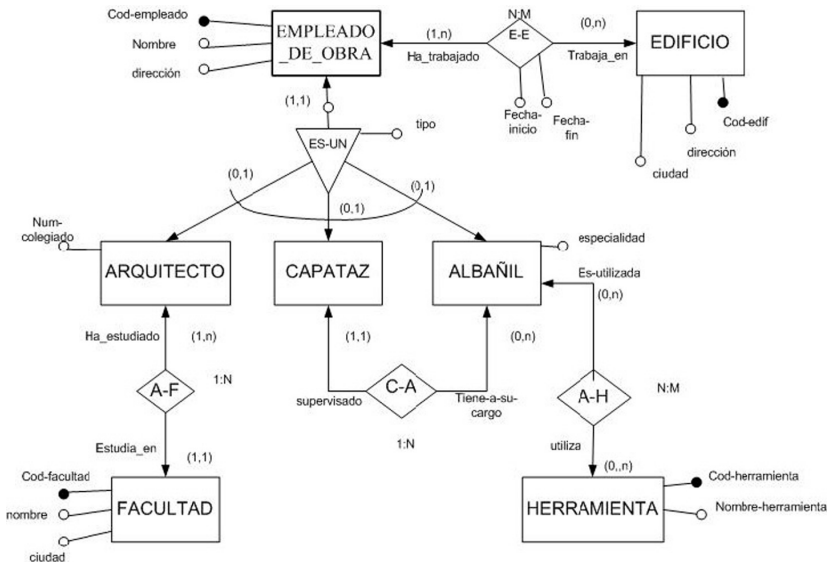
No siempre debemos aplicar una regla, pues cada regla se aplicará cuando se den una serie de circunstancias. Veamos cada una de ellas.

Eliminación del supertipo

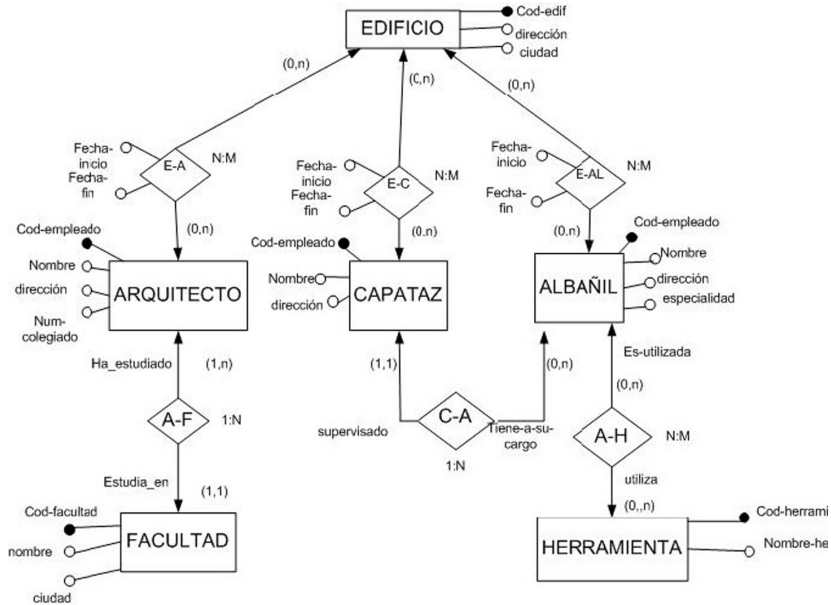
Esta regla sólo debe utilizarse cuando las *interrelaciones jerárquicas son totales y exclusivas*. Cuando se aplica todos los atributos del supertipo se transfieren a cada uno de los subtipos. Todos los tipos de interrelación que tiene el supertipo se pasarán a cada uno de los subtipos. Al ser un tipo de interrelación jerárquica exclusiva, los subtipos deberán tener cardinalidad mínima cero en los tipos de interrelación que se transfieren desde el supertipo. Los subtipos mantienen todos los tipos de interrelación que tenían.

Esta regla se deberá aplicar cuando el número de atributos transferidos sea pequeño y no existan muchos tipos de interrelación en los que participe el supertipo de entidad.

Ejemplo:



Al aplicar esta regla nos quedaría:



El esquema relacional correspondiente sería:

EDIFICIO(Cod-edificio, dirección, ciudad)

ARQUITECTO(Cod-empleado, Nombre, dirección, Num-colegiado, **Cod-facultad**)

CAPATAZ(Cod-empleado, Nombre, dirección)

ALBAÑIL(Cod-empleado, Nombre, dirección, especialidad, **Cod-empleado_capataz**)

FACULTAD(Cod-facultad, nombre, ciudad)

HERRAMIENTA(Cod-herramienta, Nombre-herramienta)

E-A(Cod-edificio, Cod-empleado, fecha-inicio, fecha-fin)

E-C(Cod-edificio, Cod-empleado, fecha-inicio, fecha-fin)

E-AL(Cod-edificio, Cod-empleado, fecha-inicio, fecha-fin)

A-H(Cod-empleado, Cod-herramienta)

Esta regla no se aplica para tipos de interrelaciones jerárquicas cuando estas son parciales, ya que se eliminaría la representación de entidades que no pertenecen a ningún subtipo, y esto sería incorrecto.

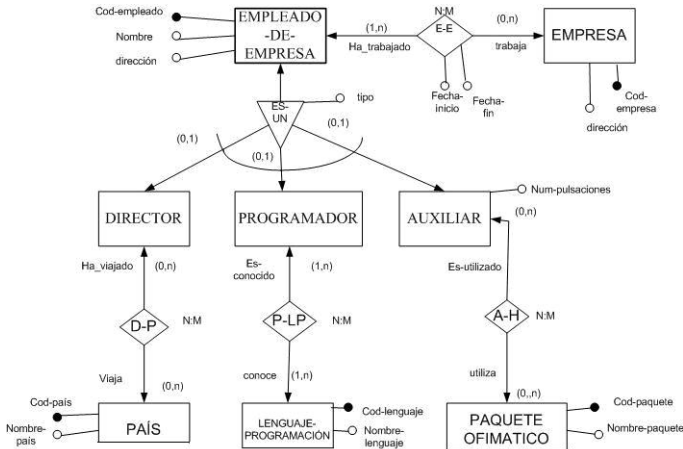
Eliminación de los subtipos

Cuando se eliminan los subtipos de entidad todos sus atributos se transfieren al supertipo, y los tipos de interrelación que tienen los subtipos se relacionarán ahora con el supertipo. El supertipo además conservará los tipos de interrelación que tuviera.

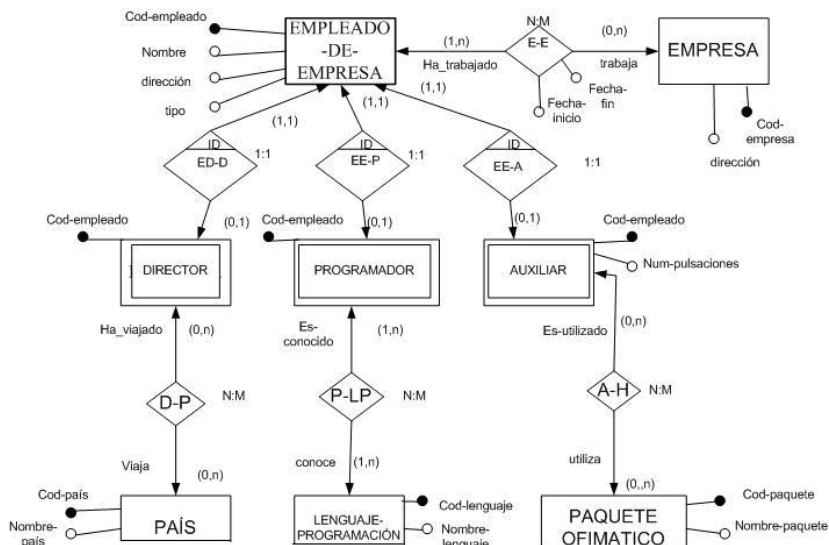
Si la jerarquía es exclusiva, el supertipo participará de forma parcial, es decir, con cardinalidad mínima cero, en los tipos de interrelación que sean transferidos desde el subtipo de entidad.

Si la jerarquía es inclusiva, participará con las cardinalidades que tenía cada uno de los subtipos en los tipos de interrelación que se transfieren.

Veamos el siguiente ejemplo:



Aplicando esta regla nos quedaría:



El esquema relacional que obtendríamos sería:

EMPRESA(Cod-empresa, dirección)

EMPLEADO-DE-EMPRESA(Cod-empleado, Nombre, dirección, tipo)

E-E(Cod-empresa, Cod empleado, fecha-inicio, fecha-fin)

DIRECTOR(Cod-empleado)

PROGRAMADOR(Cod-empleado)

AUXILIAR(Cod-empleado, num-pulsaciones)

PAÍS(Cod-país, nombre-país)

LENGUAJE-PROGRAMACIÓN(Cod-lenguaje, nombre-lenguaje)

PAQUETE-OFIMÁTICO (Cod-paquete, nombre-paquete)

D-P(Cod-empleado, Cod-país)

P-LP(Cod-empleado, Cod-lenguaje)

H-A(Cod-empleado, Cod-paquete)

6.- NORMALIZACIÓN

Después de transformar el modelo entidad-relación al modelo relacional, es preciso normalizar dicho modelo relacional aplicando una serie de reglas.

El proceso de normalización se hace necesario para:

- Evitar la redundancia de los datos y las inconsistencias.
- Evitar la incapacidad de almacenar ciertos datos.
- Evitar la ambigüedades y pérdida de información.
- Evitar problemas de actualización (anomalías de inserción, borrado y modificación) de los datos en las tablas.
- Proteger la integridad de los datos.

La teoría de la normalización fue desarrollada inicialmente por Codd, él desarrolló unas reglas llamadas *formas normales*, posteriormente Boyce y Fagin también aportaron otras reglas de normalización. Codd en 1970 definió la primera forma normal, luego se definieron la segunda, la tercera, la Boyce –Codd, la cuarta y la quinta forma normal.

Podemos definir la *Teoría de la Normalización* como la descomposición sin pérdida de información ni de semántica de la relación universal (o de una colección de relaciones equivalentes a la misma) en una colección de relaciones en la que las anomalías de actualización (inserción, borrado y modificación) no existan o sean mínimas.

Para poder entender la Teoría de Normalización debemos conocer el concepto de Dependencia Funcional.

6.1.- Concepto de dependencia funcional

Primero vamos a definir el concepto de *descriptor*:

Dada la relación $R(A_1, A_2, \dots, A_n)$, se dice que X es un descriptor de R , si $X \subseteq (A_1, A_2, \dots, A_n)$.

El concepto de *dependencia funcional (df)* lo definiremos del siguiente modo:

Sea la relación $R(A_1, A_2, \dots, A_n)$ y sean:

$X \subseteq (A_1, A_2, \dots, A_n)$ e $Y \subseteq (A_1, A_2, \dots, A_n)$ dos descriptores de R .

Se dice que Y *depende funcionalmente de* X si a cada valor de X le corresponde un único valor de Y .

$X \rightarrow Y$ (X implica o determina Y)

X es el determinante o implicante. Y es el implicado.

Ejemplo:

$\text{DNI} \rightarrow \text{Nombre-persona}$

El DNI determina funcionalmente el nombre de la persona.

Nombre-persona depende funcionalmente del DNI.

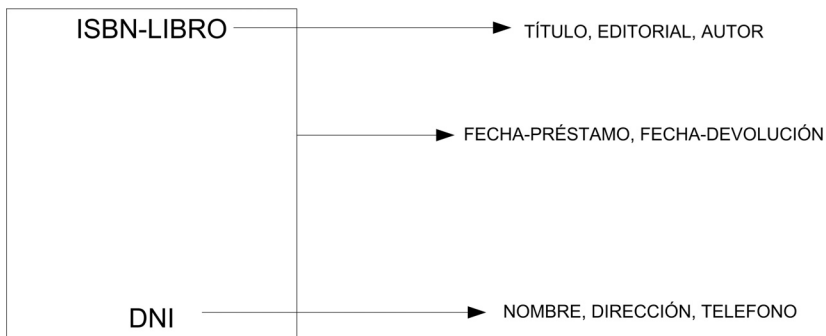
DNI es el determinante y Nombre-persona es el implicado.

Esto significa que para dos tuplas que tengan el mismo DNI, deben de tener también e mismo Nombre-Persona.

Las dependencias funcionales tienen las siguientes características:

- Son propiedades inherentes al contenido semántico de los datos, que se han de cumplir para *cualquier extensión* del esquema de relación.
- Se trata de restricciones de integridad que permiten conocer qué interrelaciones existen entre los atributos del mundo real.
- Existen otras restricciones de integridad que no se pueden expresar por medio de dependencias.
- Siempre se estudian en el ámbito de una relación.
- Son invariantes en el tiempo.

El *Grafo o Diagrama de Dependencias Funcionales* permite representar gráficamente el conjunto de atributos y las dependencias funcionales existentes entre ellos. Para poder entender este concepto veremos un ejemplo:



El ISBN-LIBRO determina funcionalmente el título del libro, la editorial y el autor.

El DNI me determina funcionalmente el nombre, la dirección y el teléfono de la persona que ha realizado el préstamo del libro.

Los atributos ISBN-LIBRO y DNI juntos me determinan la fecha del préstamo y la fecha de la devolución.

6.2.- Dependencia funcional completa

Dependencia funcional completa (DFC)

Si tenemos la DF: $X \rightarrow Y$, y el descriptor X es compuesto: $X(X1, X2)$

Y tiene dependencia funcional completa respecto de X si:

$X1 \twoheadrightarrow Y$

$X2 \twoheadrightarrow Y$

$X \rightarrow Y$

Y no depende de ningún subconjunto de X, sino que depende de todos los atributos que constituyen X. Esta dependencia funcional completa se puede representar mediante:

$$X \Rightarrow Y$$

Ejemplo:

Partiendo del grafo anterior podemos ver que ISBN-LIBRO me determina el Título, la editorial y el autor. Al estar la clave formada por dos atributos (ISBN-LIBRO, DNI), vemos que no depende de toda la clave, sino que depende de una parte de ella, no existe por ello una dependencia funcional completa para esta dependencia. Por el contrario en el caso:

$$(\text{ISBN-LIBRO, DNI}) \rightarrow \text{FECHA-PRÉTAMO}$$

Esta dependencia sí es completa, pues la fecha de préstamo del libro depende de toda la clave, el isbn del libro y el dni de la persona que realizó el préstamo. Es por tanto, una dependencia funcional completa y se puede expresar del siguiente modo:

$$(\text{ISBN-LIBRO, DNI}) \Rightarrow \text{FECHA-PRÉTAMO}$$

Dependencia Funcional Mutua

Sean X e Y, son dos descriptores de R, si se cumple que:

$$X \rightarrow Y$$

$$Y \rightarrow X$$

entonces X e Y son **equivalentes**.

$$X \ll Y$$

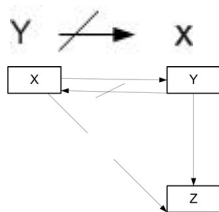
Se dice que X e Y tienen una dependencia mutua.

Dependencia Funcional Transitiva

Dada la relación R (X, Y, Z) en la que existen las siguientes dependencias funcionales:

$$X \rightarrow Y$$

$$Y \rightarrow Z$$



Se dice que Z tiene una dependencia TRANSITIVA respecto a X a través de Y, y esto se escribe mediante:

$$X \overset{Y}{\rightarrow} Z$$

Veamos un ejemplo:

Tenemos la relación PERSONA (dni, nombre, dirección, ciudad, comunidad), en la cual vemos las siguientes dependencias funcionales:

Dni \rightarrow nombre, dirección, ciudad

Ciudad \rightarrow comunidad

Y además Ciudad $\not\rightarrow$ Dni.

Vemos pues que existe una dependencia funcional transitiva entre el dni y la comunidad:

Dni \rightarrow comunidad

6.3.- Dependencia Funcional Multivaluada (DMV)

Dada una relación R con los atributos A, B y C, la dependencia multivaluada (DMV)

$$A \twoheadrightarrow B,$$

Se cumple en R, si y sólo si el conjunto de valores de B correspondiente a un par dado (valor de A, valor de C) en R, depende sólo del valor de A y es independiente del valor de C.

Como siempre los atributos A, B y C pueden ser compuestos. Igual que ocurre con las dependencias transitivas, las dependencias multivaluadas pueden existir sólo si la relación R tiene al menos tres atributos.

Se puede demostrar que dada la relación $R(A, B, C)$, la DMV $A \rightarrow B$ se cumple si y sólo si se cumple la DMV $A \rightarrow C$

Se suelen representar ambas dependencias como:
 $A \rightarrow B \frac{1}{2} C$.

Una dependencia funcional es una DMV en la cual el conjunto de valores dependientes que corresponden a un valor dado del determinante es, en realidad, un solo valor, así pues toda dependencia funcional es una DMV, pero lo inverso no es cierto.

Ejemplo:

Si tenemos una relación con los datos de Curso, Texto y Profesor en la que se cumple que “ para un Curso dado y un Texto dado, el conjunto de Profesores correspondientes depende únicamente del valor de Curso, sin importar el valor de Texto elegido”. Así Curso me multidetermina a Profesor o Profesor tiene una dependencia multivaluada de Curso. De forma análoga, “para un Curso dado y un Profesor dado, el conjunto de Textos correspondientes depende únicamente del valor de Curso sin importar el Profesor”. Vemos que Curso también multidetermina a Texto.

6.4.- Primera Forma Normal

Una relación R se encuentra en primera forma normal (1FN) si y sólo si todos los dominios simples subyacentes contienen solo valores atómicos. Es decir, que el cruce de una fila con una columna sólo tiene un dato, o sea, que no existen grupos repetitivos.

Ejemplo:

Tenemos la relación R (dni, nombre, teléfono)

Dni	Nombre	teléfono
30993309	Ana Incera	957-88-00-55 687594873
34777555	Pedro López	91-979-69-96 602998877

Como vemos existen grupos repetitivos, pues existen dos teléfonos para una misma tupla. No está en primera forma normal. Se soluciona repitiendo toda la tupla para cada uno de los valores del grupo repetitivo.

Dni	Nombre	teléfono
30993309	Ana Incera	957-88-00-55
30993309	Ana Incera	687594873
34777555	Pedro López	91-979-69-96
34777555	Pedro López	602998877

6.5.- Segunda Forma Normal

Una relación R se encuentra en segunda forma normal si y sólo si está en primera forma normal y todos los atributos no clave, dependen funcionalmente de forma completa de la clave primaria.

Sea una relación R(A,B,C,D) con clave primaria (A,B) y tal que $A \rightarrow B$. Evidentemente no está en segunda forma normal, ya que el atributo B no es clave y depende de un subcon-

junto de la clave primaria (A). Para pasar esta relación a segunda forma normal debo descomponerla en dos nuevas relaciones sin pérdida de información.

R1(A, D) siendo la clave primaria A.

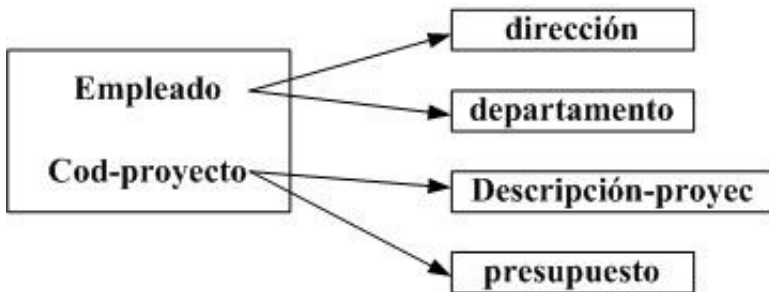
R2(A, B, C) siendo la clave primaria A,B

Estando R1 y R2 en segunda forma normal.

Ejemplo:

Tenemos la relación R(Empleado, Cod-proyecto, dirección, depto, descripción-proyec, presupuesto) con el siguiente grafo de dependencias funcionales:

Suponemos que ya está en primera forma normal, pero vemos que no está en segunda forma normal puesto que existen atributos no clave que tienen dependencia funcional no completa de la clave principal (Empleado, Cod-proyecto):



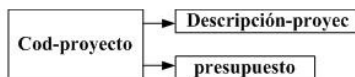
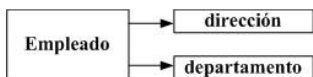
Empleado → dirección

Empleado → departamento

Cod-proyecto → descripción

Cod-proyecto → presupuesto

Para conseguir que esté en segunda forma normal debo descomponer la relación R en dos relaciones:



En estas nuevas relaciones todas las dependencias funcionales son completas, por tanto están en segunda forma normal.

6.6.- Tercera Forma Normal

Se dice que una relación está en tercera forma normal si y sólo si está en segunda forma normal y todos los atributos no clave dependen de manera no transitiva de la clave primaria.

Sea una relación $R(A,B,C)$ con clave primaria (A) tal que se dé la dependencia $B \rightarrow C$ y $A \rightarrow B$. No está en tercera forma normal, pues C no es clave y depende transitivamente de la clave primaria a través de B . Para pasar la relación R a tercera forma normal se descompone en dos nuevas relaciones sin pérdida de información.

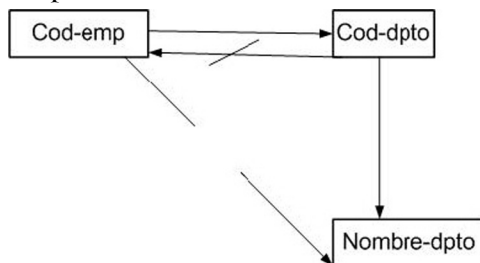
$R_1(B, C)$ siendo la clave primaria B .

$R_2(A, B)$ siendo la clave primaria A .

Estando R_1 y R_2 en tercera forma normal.

Ejemplo:

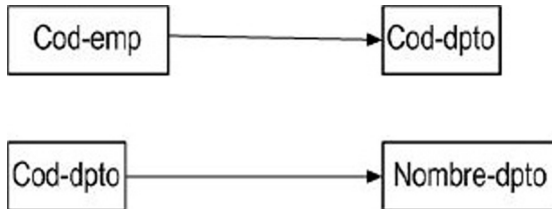
Tenemos la relación EMP-DPTO ($Cod-emp$, $Cod-dpto$, $Nombre-dpto$) que no está en tercera forma normal, ya que $Nombre-dpto$ depende transitivamente de $Cod-emp$ por medio de $Cod-dpto$.



Para pasarla a tercera forma normal se descompone en las relaciones EMPLEADO y DEPARTAMENTO que sí están en tercera forma normal.

EMPLEADO(Cod-emp, Cod-dpto)

DEPARTAMENTO(Cod-dpto, Nombre-dpto)



6.7.- Forma Normal de Boyce-Codd (FNBC)

Antes de definir la Forma Normal de Boyce-Codd, debemos introducir un término nuevo. Definimos un determinante como un atributo del cual depende funcionalmente de manera completa algún otro atributo.

Una relación está en Forma Normal de Boyce-Codd, si y sólo si está en tercera forma normal y todo determinante es una clave candidata.

Son pocos los casos de relaciones que se encuentran en tercera forma normal y no están en Forma Normal de Boyce-Codd. Ocurre cuando se presentan las siguientes circunstancias:

- Existen varias claves candidatas.
- Las claves candidatas son compuestas.
- Las claves candidatas se solapan, es decir, tienen algún atributo en común.

Sea la relación $R(A,B,C,D)$ con claves candidatas (A,B) y (B,C) y tal que $A \ll B$, evidentemente no se encuentra en forma

normal de Boyce-Codd, ya que el atributo A y C son determinantes y no son claves. Sin embargo, si se encuentra en tercera forma normal ya que no existen dependencias transitivas y además todo atributo no clave depende funcionalmente de manera completa de la clave. Para pasar la relación R a Forma Normal de Boyce-Codd se puede hacer de dos modos:

R1(A, C) siendo la clave primaria A.

R2(B, C, D) siendo la clave primaria B, C.

O bien:

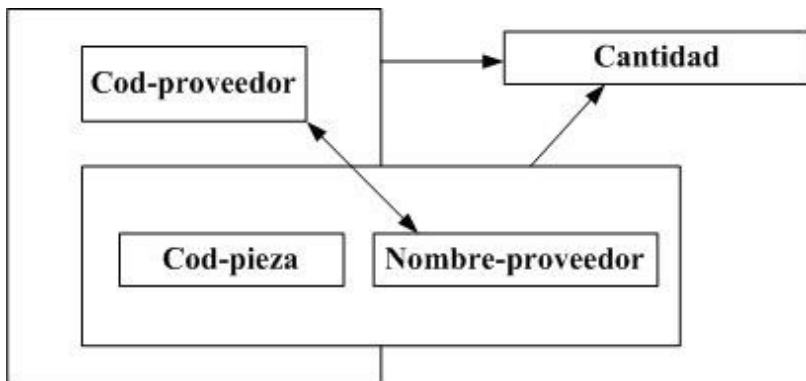
R1(A, C) siendo la clave primaria A.

R2(A, B, C) siendo la clave primaria A,B.

Estando en ambos casos R1 y R2 en Forma Normal de Boyce-Codd.

Ejemplo:

Tenemos la relación ENVIOS(Cod-proveedor, Cod-pieza, Nombre-proveedor, cantidad) con el siguiente grafo de dependencias funcionales:



Las claves candidatas son (Cod-proveedor, Cod-pieza) y (Nombre-proveedor, Cod-pieza), y además existen dos determinantes Cod-proveedor y Nombre-proveedor que no son claves candidatas.

Si la descomponemos en función de la dependencia mutua entre Cod-proveedor y Nombre-proveedor, obtenemos las siguientes relaciones:

PROVEEDORES(Cod-proveedor, Nombre-proveedor)
ENVIOS1(Cod-proveedor, Cod-pieza, cantidad)
O bien las relaciones:
PROVEEDORES(Cod-proveedor, Nombre-proveedor)
ENVIOS2(Nombre-proveedor, Cod-pieza, cantidad)

Ambos conjuntos de relaciones están en Forma Normal de Boyce-Codd.

6.8.- Cuarta Forma Normal

Una relación R se encuentra en cuarta forma normal, si y sólo si está en Forma Normal de Boyce-Codd y no existen dependencias multivaluadas.

En caso de tener una relación en Forma Normal de Boyce-Codd, y con dependencias multivaluadas, existe un teorema demostrado por **Fagin** que permite descomponer sin pérdidas la relación R deshaciendo las dependencias multivaluadas.

Teorema de Fagin

Si tenemos la relación R(A, B, C) con las siguientes dependientes multivaluadas:

A \twoheadrightarrow B
A \twoheadrightarrow C

Entonces la relación R puede descomponerse como $R_1(A, B)$ Y $R_2(A, C)$

Ejemplo:

Partiendo del ejemplo anterior, la relación $R(\text{Curso}, \text{Profesor}, \text{Texto})$ no se encuentra en cuarta forma normal porque existen las siguientes dependencias multivaluadas:

$\text{Curso} \twoheadrightarrow \text{Profesor}$

$\text{Curso} \twoheadrightarrow \text{Texto}$

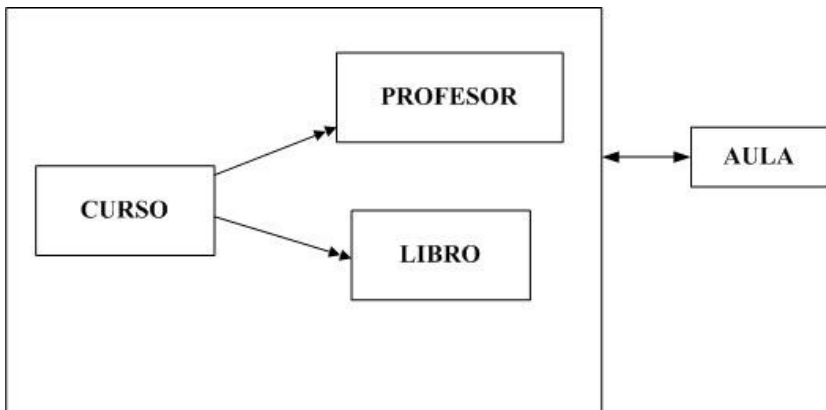
Para pasarlo a cuarta forma normal descomponemos esta relación R utilizando el Teorema de Fagin, en las siguientes relaciones:

$R_1(\text{Curso}, \text{Profesor})$

$R_2(\text{Curso}, \text{Texto})$

Veamos otro ejemplo:

Partimos del siguiente grafo de dependencias funcionales



Con la siguiente tabla R:

CURSO	PROFESOR	LIBRO	AULA
Física	Luís	A	1
Física	Luís	B	2
Física	Paco	A	3
Física	Paco	B	4
Lengua	Pepe	C	5
Lengua	Pepe	D	6
Lengua	Ana	C	7
Lengua	Ana	D	8
Lengua	Juan	C	9
Lengua	Juan	D	10

Esta relación está en Forma Normal de Boyce-Codd . Pero vemos que existen dos dependencias multivaluadas, y por tanto no está en cuarta forma normal. Descomponemos esta relación en dos relaciones:

R1(Curso, Profesor, Aula)

R2(Curso, Libro, Aula)

Añadimos el atributo Aula para no perder información. Vemos que así ya tenemos las dos relaciones en cuarta forma normal al desaparecer las dependencias multivaluadas.

Esta relación está hasta forma normal de Boyce-Codd, pero no está en cuarta forma normal, ya que existen dependencias multivaluadas:

Modelo → Color

Modelo → Equipamiento

6.9.- Quinta Forma Normal

Para definir la quinta forma normal, primero debemos definir una dependencia de reunión. Se dice que una relación R satisface la *dependencia de reunión* (DR)(X, Y,...Z) si y sólo si R es igual a la reunión de sus proyecciones según X,Y,...Z donde X,Y,...Z son subconjuntos de atributos de la relación R.

Si por ejemplo, tenemos relación SPJ que está en cuarta forma normal, ya que todos los atributos forman parte de la clave, y no existen dependencias transitivas, ni multivaluadas.

SPJ

S	P	J
S1	P1	J2
S1	P2	J1
S2	P2	J1
S1	P1	J1

Dicha relación se puede descomponer en sus proyecciones SP, PJ y JS de tal forma que al reunir dichas relaciones se obtiene la relación SPJ.

SP

S	P
S1	P1
S1	P2
S2	P1

PJ

P	J
P1	J2
P2	J1
P1	J1

JS

J	S
J2	S1
J1	S1
J1	S2

Partiendo de las relaciones SP y PJ y haciendo la reunión según P obtendremos:

S	P	J
S1	P1	J2
S1	P1	J1
S1	P2	J1
S2	P1	J2
S2	P1	J1

La tupla (S2, P1, J2) es una tupla espuria, es decir, es información incorrecta .

Si se hace una reunión entre esta última relación y la relación JS, obtendremos la relación SPJ original. Vemos que la relación SPJ tiene una dependencia de reunión respecto a sus tres posibles proyecciones, la cual se expresa como $DR^*(SP, PJ, JS)$. Por ello, podríamos descomponerla en tres relaciones sin pérdida de información.

Una relación R está en quinta forma normal, llamándose también forma normal de proyección-reunión, si y sólo si además de estar en cuarta forma normal, toda dependencia de reunión R es una consecuencia de las claves candidatas.

Para pasar una relación que no está en quinta forma normal, es decir, que contiene dependencias de reunión en las que no intervienen sus claves candidatas, será suficiente con descomponerla en las proyecciones de las que depende.

BIBLIOGRAFÍA

- DATE, C.J.: *Introducción a los Sistemas de Base de Datos*. Addison-Wesley Iberoamericana.
- KORTH, F.; SILBERSCHATZ, Henry y Abraham: *Fundamento de bases de datos (2ª Edición)*. McGraw-Hill.
- DE MIGUEL, Adoración; PIATTINI, Mario: *Concepción y Diseño de Bases de Datos*. Ra-ma. 1993.
- DE MIGUEL, Adoración; PIATTINI, Mario: *Fundamentos y modelos de Bases de Datos*. Ra-ma. 1997.
- LUQUE RUÍZ, Irene; GOMEZ-NIETO, Miguel Ángel: *Diseño y uso de Bases de Datos Relacionales*. Ra-ma. 1997.

