

Diseño lógico de datos.  
Transformación del modelo conceptual  
a modelos lógicos. Análisis relacional  
de datos. Documentación.

## **TEMA 53**

---

**ABACUS NT**

## Índice

---

- 1. Introducción**
- 2. Modelos de datos. Esquemas.**
- 3. Diseño relacional**
  - 3.1. Metodología de diseño**
- 4. Diseño conceptual: El modelo entidad/Relación**
  - 4.1. Elementos básicos**
  - 4.2. Grado de la relación**
  - 4.3. Cardinalidad de la relación (tipo de correspondencia)**
  - 4.4. Cardinalidades de las entidades**
  - 4.5. Entidades débiles vs fuertes**
- 5. Transformación del modelo conceptual a modelos lógicos**
  - 5.1. Transformación del diagrama E/R en relaciones**
    - 5.1.1. Transformación de entidades**
    - 5.1.2. Transformación de atributos de entidades**
    - 5.1.3. Transformación de relaciones**
    - 5.1.4. Transformación de atributos de relaciones**
- 6. Análisis relacional de datos. Normalización**
  - 6.1. Dependencias funcionales**
  - 6.2. Formas normales**
    - 6.2.1. Primera forma normal**
    - 6.2.2. Segunda forma normal**
    - 6.2.3. Tercera forma normal**
    - 6.2.4. Forma normal de Boyce/Codd (BCNF)**
    - 6.2.5. Cuarta forma normal**
    - 6.2.6. Quinta forma normal**
- 7. Documentación**
  - 7.1. Especificación de los requerimientos de acceso**
  - 7.2. Diccionario de datos**
- 8. Conclusión**
  - 8.1. Sistema educativo**
- 9. Bibliografía**

## 1. Introducción

Al igual que los ficheros, los sistemas de bases de datos almacenan los datos como tipos de registro. A diferencia de aquéllos, un sistema de gestión de bases de datos (**SGBD**) también almacena los enlaces entre registros. Todo SGBD soporta un tipo diferente de estructura de enlace entre tipos de registro. Así, virtualmente todo SGBD soporta una única estructura. Sin embargo, hay algunas características comunes en algunos SGBD. Estas características comunes surgen de la estandarización de los SGBD. Ahora hay varios SGBD que soportan el modelo relacional. Se conocen como **SGBD relacionales**. Hay también varios SGBD que soportan estructuras en red. Se propuso un estándar de estructuras en red, y siguiéndolo han surgido varios SGBD. Tales estructuras se conocen como **SGBD en red**. Finalmente, hay varios SGBD que soportan las estructuras jerárquicas conocidas como **SGBD jerárquicas**.

## 2. Modelos de datos. Esquemas.

Un modelo de datos es una colección de conceptos para la descripción de los datos, las relaciones entre ellos y las restricciones que deben cumplir. Un esquema es una descripción de una base de datos mediante un modelo de datos. Los modelos de datos se clasifican en:

- Modelos conceptuales. Describen los datos con un alto nivel de abstracción, utilizando entidades (concepto del mundo real), atributos (propiedad de interés de una entidad) y relaciones (interacción entre dos o más entidades). Son independientes de la base de datos a utilizar. Por ejemplo, el modelo entidad/relación y el modelo orientado a objetos.
- Modelos lógicos. Representan los datos valiéndose de estructuras de registros de varios tipos, formados por un número determinado de campos. Son dependientes de la base de datos a utilizar. Por ejemplo, el modelo relacional, el modelo de red y el modelo jerárquico.
- Modelos físicos. Los modelos físicos describen cómo se almacenan los datos en cuanto al formato de los registros, la estructura de los ficheros y los métodos de acceso utilizados.

El diseño de bases de datos se estructura en tres pasos:

- Diseño conceptual. Recibe como entrada la especificación de requerimientos y su resultado es el esquema conceptual, que es una descripción de alto nivel de la estructura de la base de datos mediante un modelo conceptual, y que es independiente del SGBD que se utilice.
- Diseño lógico. Recibe como entrada el esquema conceptual y da como resultado un esquema lógico, que es una descripción de la estructura de la base de datos mediante un modelo lógico, y que puede ser procesado por el SGBD que se utilice.
- Diseño físico. Recibe como entrada el esquema lógico y da como resultado un esquema físico, que es una descripción mediante un modelo físico de las

estructuras de almacenamiento y de los métodos usados para tener un acceso efectivo a los datos.

## 3. Diseño relacional

### 3.1. Metodología de diseño

La metodología de diseño de una BD relacional suele seguir las siguientes etapas:

- **Especificación de requisitos:** En esta fase se realiza una caracterización completa de las **necesidades** de datos de los posibles usuarios de la BD.
- **Diseño conceptual:** En esta fase se traduce los requisitos en un esquema conceptual (el cual es independiente del SGBD). Se suele emplear el modelo entidad-relación (E/R).
- **Especificación de requisitos funcionales:** En esta fase los usuarios describen los tipos de operaciones (transacciones) que se llevarán a cabo sobre los datos.
- **Diseño lógico:** se traduce el esquema conceptual de alto nivel (normalmente un diagrama E/R) al **modelo de datos** de la implementación del SGBD que se va a usar (normalmente el modelo relacional).

## 4. Diseño conceptual: El modelo entidad/Relación

El modelo E/R fue propuesto por **Peter P. Chen**, en los años 70. Presenta una vista unificada de los datos, centrándose en la estructura lógica y abstracta de los mismos, como representación del mundo real.

Inicialmente (en la propuesta de Chen) sólo se incluían los conceptos de entidad, relación y atributos. Después se añadieron otras propuestas (atributos compuestos, generalizaciones...) que forman el llamado **modelo entidad relación extendido**.

### 4.1. Elementos básicos

Los elementos con los que se construye un diagrama E/R son:

**Entidad:** Es un objeto del que nos interesa almacenar información, por ejemplo, puede ser una persona. Se representa mediante un rectángulo.

**Relaciones:** Se define como la asociación entre dos o más entidades. Se representa con un líneas, rombo, e indicando el **tipo de correspondencia**, la cual puede ser 1:1 o 1:N, por ejemplo.

**Atributos:** Es cada una de las propiedades que tiene un tipo de entidad o relación, por ejemplo, nombre, o precio. Se representa mediante una elipse o un círculo. Aquellos que son utilizados para identificar únicamente las instancias de una entidad (las claves) se representan sombreados o subrayados.

## 4.2. Grado de la relación

Es el número de entidades que participan en una relación. Normalmente las relaciones son **binarias**, pero también podrían ser **ternarias** o de n miembros (**n-arias**). Existe un grado especial que es el de las relaciones **unarias**, que se utiliza para relaciones reflexivas de una entidad consigo misma.

## 4.3. Cardinalidad de la relación (tipo de correspondencia)

Es el número máximo de instancias de cada entidad que pueden intervenir. También recibe el nombre de “*tipo de correspondencia*”. En la representación gráfica aparece como una etiqueta con 1:1, 1:n, n:1 o n:m, que se leen respectivamente como uno a uno, uno a muchos, muchos a uno y muchos a muchos.

**Cardinalidad 1:1 (uno a uno):** cuando en la interrelación sólo puede aparecer, como máximo, una ocurrencia del tipo de entidad por cada ocurrencia del otro.

**Cardinalidad 1:n (uno a muchos):** si para uno de los tipos de entidad puede haber un número indefinido (mayor que uno) de ocurrencias

**Cardinalidad n:m (muchos a muchos):** Si para cada ocurrencia de un tipo de entidad hay un número indefinido y mayor que uno de ocurrencias, y viceversa para el otro tipo de entidad.

## 4.4. Cardinalidades de las entidades

Se define como el número máximo y mínimo de ocurrencias de una entidad que participan en el tipo de relación.

## 4.5. Entidades débiles vs fuertes

Al igual que las entidades, las relaciones se clasifican también en:

**Fuertes.** Asocian dos entidades fuertes.

**Débiles.** Asocian una entidad débil con otra fuerte.

## 5. Transformación del modelo conceptual a modelos lógicos

El diseño lógico consiste **en transformar el modelo de datos conceptual en un modelo lógico para la base de datos**.

Una vez transformado, es necesario verificar que el modelo generado tiene un mínimo de calidad. Para ello aplicamos la teoría de la **normalización** que nos guiará en el proceso de optimización del diseño lógico.

## 5.1. Transformación del diagrama E/R en relaciones

El paso de un esquema en el modelo E/R al relacional está basado en los siguientes principios:

- Todo tipo de entidad se transforma en una relación
- Todo tipo de relación N:M se transforma en una relación
- Todo tipo de relación 1:N se traduce en el fenómeno de propagación de clave del lado de entidad 1 al lado N o bien se crea una nueva relación.

Durante este paso se perderá algo de semántica, pero esto no va a afectar en absoluto, ya que las restricciones de integridad referencial se asegurarán de la conservación de esta.

### 5.1.1. Transformación de entidades

Todas las entidades se pararán a tabla.

### 5.1.2. Transformación de atributos de entidades

Cada atributo de la entidad se transforma en una columna de la tabla, con un tipo de dato acorde a los valores que puede tomar. Si el atributo está subrayado, es decir, es clave principal, debe reflejarse esto en el diseño de la tabla.

### 5.1.3. Transformación de relaciones

La transformación dependerá de la cardinalidad de la relación:

**N:M** Se transforma en una relación entre las dos entidades (ahora tablas) en la que la clave primaria de una debe ser clave ajena en la otra y viceversa. Es necesario considerar los casos de borrado o actualización de la clave primaria.

**1:N** La propagación de los atributos principales de la entidad que tiene cardinalidad máxima 1 a la que tiene cardinalidad máxima N, es decir del lado a 1 al N, se aplica cuando la cardinalidad es obligatoria es decir (1,1) y (0,n) o (1,n). Desaparece el nombre de la relación, y si existen atributos propios de ésta también se propagan.

Cuando la cardinalidad sea (0,1) y (0,n), se construye una tabla con la relación, siendo la clave de esta tabla la de la entidad del lado N (muchos).

**1:1:** Por lo que respecta a las interrelaciones 1:1, se pueden recoger en el esquema relacional de distintas maneras atendiendo a las cardinalidades de las entidades que participan en la misma: Si las entidades que se asocian poseen cardinalidades (0,1), en este caso la interrelación 1:1 se transformará en una relación, además de las dos relaciones que representan cada una de las entidades. Si una de las entidades que participa en la interrelación posee cardinalidades (0,1), mientras que en la otra son (1,1), conviene propagar la clave de la entidad con cardinalidades (1,1) a la tabla resultante de la entidad de cardinalidades (0,1). En el caso de que ambas entidades presenten cardinalidades (1,1),

se puede propagar la clave de cualquiera de ellas a la tabla resultante de la otra, teniendo en cuenta en este caso, los accesos más frecuentes y prioritarios a los datos de las tablas. Se puede plantear -también por motivos de eficiencia- la propagación de las dos claves. Sin embargo, esto introduce redundancias que deben ser controladas por medio de restricciones.

### Relaciones reflexivas (unarias):

Si la relación es N:M se trata igual que en las relaciones binarias.

Si la relación es 1:N existen 2 opciones:

- Entidad muchos es obligatoria se procede como en el caso 1:1.
- Entidad muchos no es obligatoria, se crea una nueva tabla cuya clave será la de la entidad del lado muchos, y además se propaga la clave como clave ajena a la nueva tabla.

Si la relación es 1:1, la clave de la entidad se repite, siendo una clave primaria, y la otra clave ajena 5.4

### Interrelaciones N-arias

En este tipo de relaciones se agrupan 3 o más entidades, cada entidad se convierte en una tabla, así como la relación, que va a contener los atributos propios, así como los pertenecientes a las claves principales de todas las entidades. La clave de la nueva tabla La clave de la nueva tabla será la concatenación de todas las claves aportadas por entidades con cardinalidades máximas N.

#### 5.1.4. Transformación de atributos de relaciones

Si la relación se transforma en una tabla, todos sus atributos pasan a ser columnas de la tabla. En el caso en que alguno de los atributos de la relación sea principal, deberá ser incluido como parte de la clave primaria en dicha tabla.

## 6. Análisis relacional de datos. Normalización

Con objeto de verificar si un determinado diseño de una BD es correcto, se aplica la teoría de la normalización, la cual veremos en detalle a continuación.

### 6.1. Dependencias funcionales

Para entender la definición de las formas normales, previamente es necesario comprender el concepto de dependencias funcionales:

Sea el **esquema de relación** R definido sobre el conjunto de atributos A y sean X e Y subconjuntos de A llamados descriptores. Se dice que, Y depende funcionalmente de X, si, y sólo si, cada valor de X tiene asociado en todo momento un único valor de Y. Se representa como  $X \rightarrow Y$

En entornos profesionales, se considera que se tiene un buen diseño si se encuentra en 3FN, no obstante, como indica el autor C.J. Date, en su libro *Introduction to Database Systems* (2003), existen aún más formas normales, como, por ejemplo, la forma normal de dominio-clave.

## 6.2. Formas normales

Las formas normales son las líneas maestras para el diseño de los registros, estas reglas están pensadas para evitar anomalías en la puesta al día e inconsistencias en los datos, A pesar de ello, no hay obligación de seguir todas las normalizaciones hasta el final.

Antes de comenzar a enunciarlas, también hay que dar unos conceptos previos:

**Dependencia funcional.** Dada una relación R, se dice que el atributo Y de R es funcionalmente dependiente del atributo X de R ( Se escribe  $R.X \rightarrow R.Y$  y se dice "R.X determina funcionalmente a R.Y") si y solo si para cada valor de X hay asociado precisamente un único valor a la vez de Y. Tanto X como Y pueden ser compuestos, es decir, grupos de atributos. Informalmente, se puede decir que Y depende de X si no puede existir dos tuplas con el mismo valor en X y distinto en Y. *Ejemplo:* El atributo *nombre* depende funcionalmente del atributo *DNI*.

**Determinante** es un atributo de una relación del que otro atributo cualquiera de ella es totalmente dependiente.

**Hecho multivaluado** es una relación muchos a muchos, como la relación hombre-empleo, en que un hombre puede tener varios empleos y un empleo (p.e. Electricista) lo pueden tener muchos hombres.

### 6.2.1. Primera forma normal

La primera forma normal simplemente define las tablas que forman el modelo relacional y excluye campos repetidos y grupos.

**Una relación está en la primera forma normal si y sólo si los dominios de sus atributos toman sólo valores atómicos. Es decir, si no contiene atributos multivaluados.**

Esta definición es equivalente a la anterior de relación normalizada e indica que una relación está en primera forma normal si de hecho es correcta.

### 6.2.2. Segunda forma normal

La segunda forma normal trata de las relaciones entre los atributos clave y los no clave. Esta forma es violada cuando un campo no clave es de hecho un subconjunto de una clave.

**Una relación R está en segunda forma normal si y sólo si está en primera forma normal y todo atributo no clave es completamente dependiente de la clave primaria. Es decir, si todos sus atributos dependen funcionalmente de toda la clave primaria. (Si la clave de una relación es simple ya se tiene la segunda forma normal).**

La relación anterior RI, viola esta regla, ya que, el atributo direcc-almacén no depende de la clave (pieza, almacén), sino que es función sólo de almacén. Para que se cumpla la segunda forma normal en este caso habrá que descomponer la relación en dos:

### 6.2.3. Tercera forma normal

La tercera forma normal no se cumple cuando un campo no clave es de hecho clave de otro campo.

La tercera forma normal se puede enunciar como:

**Una relación está en la tercera forma normal si y sólo si está en la segunda y todo atributo no-clave es dependiente de la clave de forma no transitiva. Es decir, si está en la segunda forma normal y ningún atributo depende transitivamente de la clave primaria.**

Este enunciado significa que la dependencia de Y con respecto a la clave X no se hace a través de un tercer atributo Z sino directamente:

$R.X \rightarrow R.Y$  en vez de  $R.X \rightarrow R.Z \rightarrow R.Y$

### 6.2.4. Forma normal de Boyce/Codd (BCNF)

La anterior definición de Codd de la tercera forma normal, tiene ciertas inadecuaciones, en concreto no trata los casos en que hay varias claves candidatas que se componen de varios atributos y están solapadas (tienen atributos comunes). Para complementarla se enuncia la forma normal de Boyce/Codd que mejora la primera definición.

*Una relación está en la forma normal de Boyce/Codd si y sólo si todo determinante es clave candidata.* Es decir, si todas las partes izquierdas de las dependencias son claves candidatas.

Toda relación en forma normal de Boyce/Codd está en la tercera forma normal por definición. La interpretación es que todo atributo que determine únicamente otro en una relación debe poder ser clave de la misma.

### 6.2.5. Cuarta forma normal

Tanto la cuarta como la quinta forma normal trabajan con hechos multivaluados.

Bajo la cuarta forma normal, dos o más hechos independientes multivaluados no pueden coexistir en la misma relación.

**Una relación está en cuarta forma normal, si y sólo si siempre que existe un hecho multivaluado,  $A \rightarrow B$ , entonces todos los atributos de la relación son funcionalmente dependientes de A. Es decir, si está en BCNF y no existe entre sus atributos ninguna dependencia multivaluada que no sea combinación de dependencias funcionales.**

Por supuesto, A y B pueden ser compuestos.

En otras palabras, las únicas dependencias (simples o multivaluadas) dentro de la relación son las del tipo  $K \rightarrow X$  donde  $K$  es una clave candidata completa y  $X$  cualquier otro atributo.

### 6.2.6. Quinta forma normal

La quinta forma normal trata los casos en que la información está en distintas piezas que hay que mantener separadamente y sin redundancia.

Informalmente, podemos decir que una relación está en la quinta forma normal cuando su información no puede ser reconstruida si la dividimos en pequeños trozos.

**Una relación está en la quinta forma normal si y sólo si toda dependencia en ella es una consecuencia de sus claves candidatas.**

## 7. Documentación

La transformación a modelos lógicos es una parte de la especificación de base de datos. Sirve para definir la estructura de la base de datos. Para el diseño de la base de datos se necesita información adicional, en particular:

- Datos cuantitativos que informan del volumen de información a almacenar.
- Requerimientos de acceso, que informan de cómo se usa la base de datos. Estos requerimientos se utilizan para elegir las claves de ficheros durante el diseño de la base de datos.

La información sobre datos cuantitativos y requerimientos de acceso se obtienen durante el análisis de sistema. Los datos cuantitativos consisten en:

- El tamaño de los elementos de datos, y
- El número de ocurrencias de tipos de registro.

El tamaño de los elementos de datos se da normalmente en el **diccionario de datos**. Los volúmenes de registro se pueden añadir a la estructura lógica de registro.

### 7.1. Especificación de los requerimientos de acceso

Los requerimientos de acceso se toman inicialmente de las especificaciones de procedimientos de usuario, que incluye sentencias sobre cómo acceden los usuarios a los datos. Estas sentencias pertenecen a los requerimientos de acceso. Durante el diseño de base de datos, se delinean las rutas de acceso a los tipos de registro lógico para cada requerimiento. Estas rutas muestran cómo se usa el dato y describen:

- Los tipos de registro accedidos por cada requerimiento de acceso.
- La secuencia en la que se accede a esos tipos de registro.
- Las claves de acceso utilizadas para seleccionar los tipos de registro.
- Los elementos recuperados de cada registro, y
- El número de registros a los que se accede.

La definición de requerimientos de acceso completa es la especificación de la base de datos, que se utiliza ahora para crear un diseño de base de datos. La estructura de registro

lógico se convierte en un **diseño lógico de base de datos**. Las rutas de acceso se utilizan para seleccionar una estructura física apropiada. Normalmente, para satisfacer los requerimientos de acceso se puede elegir varias alternativas físicas. Parte del diseño es seleccionar a partir de estas estructuras. Para hacer esas elecciones se utiliza un proceso iterativo que estima los tratamientos de cada iteración para comparar las alternativas.

Las clases de técnicas de diseño que se usen dependerán del software que se utilice para la implementación de la base de datos. La conversión más simple es ir hacia un **conjunto de ficheros**. Alternativamente, la base de datos se puede implementar en un **SGBD**. Primero se describirá la conversión en un conjunto de ficheros y después en SGBD. También se hará un breve estudio de técnicas de diseño utilizadas para convertir una estructura lógica de registro en una estructura de base de datos.

## 7.2. Diccionario de datos

Debe existir una verdadera arquitectura de datos que facilite la integración. El núcleo de dicha arquitectura será el **diccionario de recursos de información (iRD)**, que contendrá las descripciones de todos los datos que constituyen el sistema de información.

Las finalidades principales de los DRI son precisamente ayudar a la gestión de la información como recurso y conseguir la integración de la semántica de las aplicaciones de forma centralizada, con lo que se aumentan los beneficios estratégicos, operacionales y de control de las empresas.

Para Codd (1990), uno de los requisitos para que un SGBD pueda considerarse como verdaderamente relacional es que soporte un catálogo dinámico en línea en el que la descripción de la base de datos se represente como cualquier otro dato, permitiendo a los usuarios autorizados aplicar el mismo lenguaje relacional tanto a la descripción de la base de datos como a los datos regulares.

Los recursos informáticos de las instituciones se gestionan almacenando, administrando y controlando los denominados **metadatos**, esto es, los datos que definen y describen los datos de la institución. El término más difundido para estos metadatos es el de **Diccionario de Datos (DD)** o catálogo del sistema.

El directorio de datos ha sido desde siempre un componente del SGBD, encargado de describir dónde y cómo se almacenan los datos de la base, el modo de acceso y otras características físicas de los mismos, atendiendo de este modo las peticiones de los programas. Contiene, en definitiva, las especificaciones necesarias para pasar de la representación externa de los datos a la representación interna de los mismos. Su objetivo principal es transmitir al SGBD la información necesaria para poder acceder a los datos contenidos en la base.

Cuando los SGBD eran más limitados, los usuarios empleaban una lista "manual" donde anotaban la descripción narrativa y técnica (tamaño, tipo...) de los datos, así como los permisos de acceso a cada uno de ellos, el uso que las aplicaciones hacían de ellos, etc. Esta lista se denominaba diccionario de datos. Consistía en información sobre los datos, o, en otras palabras, datos sobre los datos, también conocidos como metadatos.

Por tanto, el diccionario de datos se considera como un conjunto de metadatos (datos sobre los datos) que contiene las características lógicas de los datos que se van a utilizar en el sistema que se programa, incluyendo nombre, descripción, alias, contenido y organización.

Si el diccionario de datos lo usan sólo los administradores, usuarios y diseñadores, y no el software del SGBD.

Por último, a partir de los 80, el diccionario de datos se incorporó a los SGBD como un componente más e integrándose junto al directorio. Al conjunto se le suene denominar catálogo o repositorio. El catálogo es el responsable de establecer la correspondencia entre los tres niveles de abstracción. Los módulos del SGBD usan y leen el catálogo con mucha frecuencia; por ello es importante implementar su acceso de la forma más eficiente posible.

Con la amplia difusión del modelo relacional el catálogo se almacena junto al resto de los datos de la base, en forma de tablas. La información almacenada en el catálogo de un SGBD relacional incluye, entre otros aspectos:

- Descripciones de los nombres de las relaciones (tablas).
- Nombres de los atributos.
- Tipos de datos de los atributos (dominios).
- Claves primarias.
- Claves ajenas.
- Obligación de establecer los datos como no nulos en algunas columnas.
- Vistas.
- Índices.
- Estructuras de almacenamiento.
- Permisos.
- Información sobre el propietario de cada tabla.

Dado que los datos del catálogo se almacenan en forma de tablas, se puede acceder a los mismos mediante instrucciones de SQL.

**Directorio de datos:** Es un subsistema del SGBD, encargado de describir dónde y cómo se almacenan los datos de la base, el modo de acceso y otras características físicas de los datos, atendiendo de este modo las peticiones de los programas y procesos. Un directorio de datos contiene, en definitiva, las especificaciones necesarias para pasar de la representación externa de los datos a su representación interna. Ha de estar siempre en un formato legible para la máquina. Su objetivo principal es transmitir al sistema la información necesaria para poder acceder a los datos contenidos en la base. El directorio, a diferencia del diccionario, es un instrumento del SGBD, y está orientado a facilitar a éste la información que necesita para su funcionamiento.

Un desarrollo importante en los SGBD relacionales es la práctica común de almacenar el directorio como un conjunto de relaciones. Esto permite el uso de los lenguajes de manipulación de datos de los SGBD para consultar, actualizar y mantener el diccionario de datos.

Recientemente, ha aparecido un nuevo concepto, el **diccionario de recursos de información (dri)**, que pretende ser el eslabón final en la evolución de los almacenes de

datos. El DRI constituye el depósito integrado de todos los datos sobre la organización, automatizados o no, que son utilizados para efectuar las labores de planificación, control y operación que permitan a la empresa cumplir sus objetivos. Éstos engloban de algún modo las capacidades y funciones de todos los almacenes de datos anteriores.

## 8. Conclusión

En este tema hemos visto el diseño de una base de datos partiendo del diagrama de E/R extendido, hasta completar el diseño lógico normalizado junto con la documentación.

Para que la base de datos sea funcional, es necesario que verifique un cierto grado de normalización.

En entornos profesionales, se considera que se tiene un buen diseño si se encuentra en 3FN, no obstante, como indica el autor C.J. Date, en su libro *Introduction to Database Systems* (2003), existen aún más formas normales, como, por ejemplo, la forma normal de dominio-clave.

Por último, el diseño de la base de datos es necesario implantarlo físicamente en un DBMS concreto. Para ello es conveniente hacer algunas medidas y provisiones sobre el funcionamiento de la BD.

### 8.1. Sistema educativo

Este tema es aplicado en el aula en los módulos profesionales siguientes, con las atribuciones docentes indicadas (PES/SAI):

#### Formación profesional básica

- Operaciones auxiliares para la configuración y la explotación (TPB en Informática de Oficina/ TPB en informática y Comunicaciones) (PES/SAI)
- Ofimática y archivo de documentos (TPB en Informática de Oficina) (PES/SAI)

#### Grado Medio

- Aplicaciones ofimáticas (GM de SMR) (PES/SAI)

#### Grado Superior

- Gestión de bases de datos (ASIR) (PES)
- Bases de Datos (DAW/DAM) (PES)
- Entornos de Desarrollo (DAW/DAM) (PES)

#### Bachillerato:

- 4º ESO – Tecnología de la Información y la comunicación (PES)
- Bachillerato – Tecnologías de la Información y la Comunicación (PES)

## 9. Bibliografía

- C.J. Date: **Introducción a los sistemas de bases de datos** Pearson, 2001.
- Elmasri, R.A. y Navathe S.B: "**Fundamentos de Sistemas de Bases de Datos**". Addison-Wesley, 3<sup>a</sup> Edic, 2002.
- Olga Pons, J M Medina, M.A. Vila. **Introducción a los Sistemas de Bases de Datos** Edt Paraninfo (2005)
- Korth, H.F. y Silberschatz: "**Fundamentos de Bases de Datos**". McGraw -Hill, 4<sup>a</sup> Edic., 2002.
- Garcia-Molina, H.; Ullman, J.D.; Widom, J. **Database systems: the complete book** - Pearson Education Limited, 2013.
- Abraham Silberschatz, Henry F. Korth, y S. Sudarshan, **Fundamentos de bases de datos** Edt. Mc Graw-Hill (2014)
- Pressman, R. S. **Ingeniería del Software. Un enfoque práctico**, 3<sup>a</sup> edición. Ed. McGraw-Hill, 2000.
- Sommerville, I.: **Ingeniería de Software**. 6<sup>a</sup> Edición. Addison-Wesley Iberoamericana, 2002.
- Piattini, M. y otros: **Análisis y diseño detallado de Aplicaciones Informáticas de Gestión**. RA-MA, 1996.
- Cerrada, J. A.: **Introducción a la Ingeniería del Software**. 1<sup>a</sup> Edición de 2000. Editorial Centro de Estudios Ramón Areces, S.A.