

Tema 40. Diseño de base de datos relacionales

## **TEMA 40**

---

**ABACUS NT**

## **Índice**

---

- 1. Introducción.**
- 2. Modelos de datos. Esquemas. Diseño de bases de datos.**
- 3. Diseño de base de datos relacionales**
  - 3.1. Metodología de diseño**
- 4. Diseño conceptual: El modelo entidad/Relación**
  - 4.1. Elementos básicos**
  - 4.2. Entidades**
  - 4.3. Atributos, dominios y claves.**
  - 4.4. Relación**
  - 4.5. Grado de la relación**
  - 4.6. Cardinalidad de la relación (tipo de correspondencia)**
  - 4.7. Cardinalidades de las entidades**
  - 4.8. Entidades débiles vs fuertes**
- 5. Modelo E/R extendido**
  - 5.1. Cardinalidades de un tipo de entidad:**
  - 5.2. Relaciones exclusivas:**
  - 5.3. Dependencias débiles:**
  - 5.4. Generalización y herencia:**
  - 5.5. Relaciones ternarias,**
  - 5.6. Construcción de un esquema E/R**
- 6. Diseño lógico**
  - 6.1. Transformación del diagrama E/R en relaciones**
    - 6.1.1. Transformación de entidades**
    - 6.1.2. Transformación de atributos de entidades**
    - 6.1.3. Transformación de relaciones**
    - 6.1.4. Transformación de atributos de relaciones**
- 7. Análisis relacional de datos. Normalización**
  - 7.1. Dependencias funcionales**
- 8. Formas normales**
  - 8.1. A. Primera forma normal**
  - 8.2. B. Segunda forma normal**
  - 8.3. C. Tercera forma normal**
  - 8.4. D. Forma normal de Boyce/Codd (BCNF)**
  - 8.5. E. Cuarta forma normal**
  - 8.6. F. Quinta forma normal**
- 9. Diseño físico**
  - 9.1. Buenas prácticas en el diseño físico**
- 10. Conclusión**
  - 10.1. Sistema educativo**
- 11. Bibliografía.**

## 1. Introducción.

Gracias al modelo conceptual **Entidad-Relación**, creado por **Peter Chen** en los años setenta, podemos representar el mundo real a través de una serie de símbolos y expresiones determinados. El objetivo de este modelo es simplificar el diseño de bases de datos partiendo de las descripciones textuales de la realidad, que establecen los requerimientos del sistema, es decir, obtener una representación de la realidad que capture las propiedades de la misma. Esta representación debe suponer una imagen fiel del comportamiento del mundo real.

Al tratarse de un modelo conceptual, éste no está orientado a ningún sistema físico concreto: tipo de ordenador, SGBD, sistema operativo... Tampoco tiene un objetivo informático claro, podría utilizarse para explicarle a un empleado cualquiera el funcionamiento de cualquier proceso de una forma natural y sencilla, por lo que debe ser un sistema de fácil comprensión para personas sin conocimientos informáticos. Además, las características actuales de este modelo favorecen la representación de cualquier tipo de sistema y a cualquier nivel de abstracción o refinamiento, lo cual da lugar a que se aplique tanto a la representación de problemas que vayan a ser tratados mediante un sistema informatizado como manual.

## 2. Modelos de datos. Esquemas. Diseño de bases de datos.

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 de base de datos relacionales

### 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. Entidades

Una entidad, representa cualquier persona, suceso, evento o concepto **sobre el que queramos almacenar información**.

La **representación gráfica** de un tipo de entidad es un rectángulo etiquetado con el nombre del tipo de entidad:



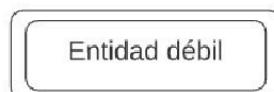
Una entidad cumple las siguientes propiedades según Tardieu:

1. Tiene existencia propia. Es decir, desde el punto de vista en el cual se estudia el sistema y al nivel de abstracción en que es considerado, la entidad existe como un elemento que interviene en el comportamiento global del sistema.
2. Es diferente del resto de las entidades (objetos) que intervienen en el sistema de información.
3. Todas las ocurrencias de un tipo de entidad deben tener los mismos tipos de características (atributos).

Las entidades las podemos clasificar en:

- **Entidades Fuertes, o regulares:** tienen existencia por sí mismas.
- **Entidades Débiles:** Su existencia depende de la existencia de otra entidad.

Las entidades débiles se representan mediante dos rectángulos concéntricos con su nombre en el interior:

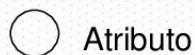


#### 4.3. Atributos, dominios y claves.

Cada una de las propiedades o características que tiene un tipo de entidad o un tipo de relación se denomina **atributo**. La representación gráfica de un atributo consiste en una elipse con el nombre del atributo en su interior.



Otra forma de representarlo, puede ser mediante un círculo y una etiqueta:



El conjunto de posibles valores que puede tomar un atributo recibe el nombre de dominio. El **dominio** tiene un nombre y una existencia propia con independencia de cualquier entidad o atributo.

Del conjunto de los atributos de una entidad, debemos elegir uno (o varios) que actúen como clave primaria. Estos atributos se representarán siempre con el nombre subrayado:

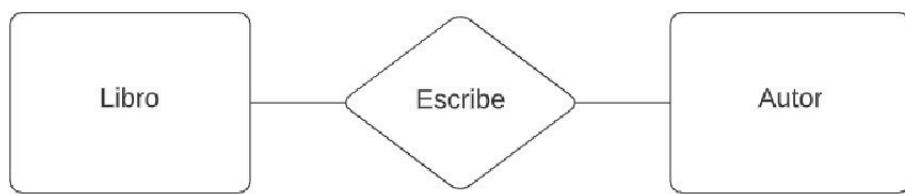


#### 4.4. Relación

Se entiende por relación aquella *asociación o correspondencia existente entre entidades*.

Se llama tipo de relación a la estructura genérica del conjunto de relaciones existentes entre dos o más tipos de entidad.

El tipo de relación se representa mediante un rombo etiquetado con el nombre de la misma, unido mediante arcos a los tipos de entidad que asocia.



Para definir una relación debemos tener en cuenta los siguientes elementos:

**Nombre de la relación:** Nombre de la relación: cada relación tiene un nombre que la distingue claramente del resto y mediante el cual ha de ser referenciada. Habitualmente se utiliza un verbo en forma singular

**Grado de la relación:** Es el número de entidades que participan en una relación.

**Cardinalidad de la relación:** Es el número máximo de instancias de cada entidad que pueden intervenir.

**Cardinalidades de las entidades:** para el modelo E/R extendido.

Veamos con detalle, estas propiedades y alguna más:

#### 4.5. 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.

#### 4.6. 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.

#### 4.7. Cardinalidades de las entidades

Esta propiedad se corresponde al modelo E/R extendido como veremos más adelante.

#### 4.8. 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. Modelo E/R extendido

Varios autores han realizado diversas extensiones al modelo E/R definido por Chen, profundizando en aspectos que aportan aún más significado y relevancia a esta herramienta para el diseño conceptual de la BD.

El potencial expresivo del modelo E/R ha sido ampliado con nuevos elementos tales como:

#### 5.1. Cardinalidades de un tipo de entidad:

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

Las cardinalidades de las entidades se definen como el número máximo y mínimo de instancias de una entidad que pueden estar relacionadas con una instancia de otra u otras entidades que participan en la relación. Su representación gráfica es una etiqueta del tipo (0,1), (1,1), (0,N) o (1,N), según corresponda. El significado de cada una es:

**(0,1)** Para una ocurrencia determinada de los otros tipos de entidades que participan en la relación, el valor mínimo de ocurrencias de la entidad que se trata es 0, y el número máximo es 1

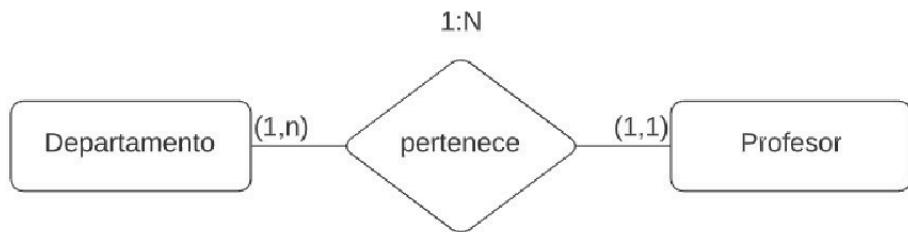
**(1,1)** Para una ocurrencia determinada de los otros tipos de entidades que participan en la relación, debe existir **una y sólo una** ocurrencia de la entidad que se trata.

**(0,n)** Para una ocurrencia determinada de los otros tipos de entidades que participan en la relación, el valor mínimo de ocurrencias de la entidad que se trata es 0, y el número máximo es n.

**(1,n)** Para una ocurrencia determinada de los otros tipos de entidades que participan en la relación, debe existir **como mínimo una** ocurrencia de la entidad que se trata, y el número máximo es n.

Aunque en este momento parezca que los conceptos "cardinalidad de una relación" y "cardinalidad de una entidad" son muy similares, su existencia se hace necesaria para la transformación del diagrama E/R al modelo relacional.

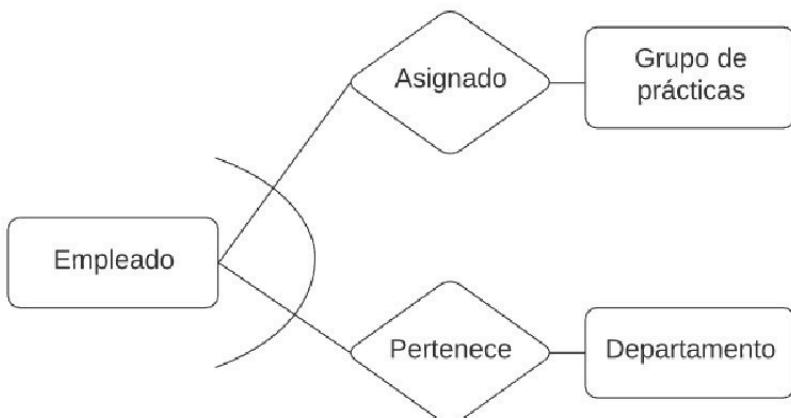
Ejemplo:



En el ejemplo, un profesor pertenece a un único departamento, mientras que un departamento está formado por mínimo un profesor y máximo n. La relación “pertenece” es de uno a muchos.

### 5.2. Relaciones exclusivas:

Cuando cada ocurrencia de un tipo de entidad sólo puede pertenecer a un tipo de relación y sólo a uno simultáneamente. Esto se representa mediante un semicírculo:



### 5.3. Dependencias débiles:

**Dependencia en existencia:** Se produce cuando una entidad (entidad débil) no pueden existir sin la ocurrencia de la entidad de la que dependen (entidad regular).

**Dependencia en identificación:** Se da cuando, además de la dependencia en existencia, las ocurrencias del tipo de entidad débil necesitan añadir la clave de la ocurrencia de la entidad regular para identificarse.

### 5.4. Generalización y herencia:

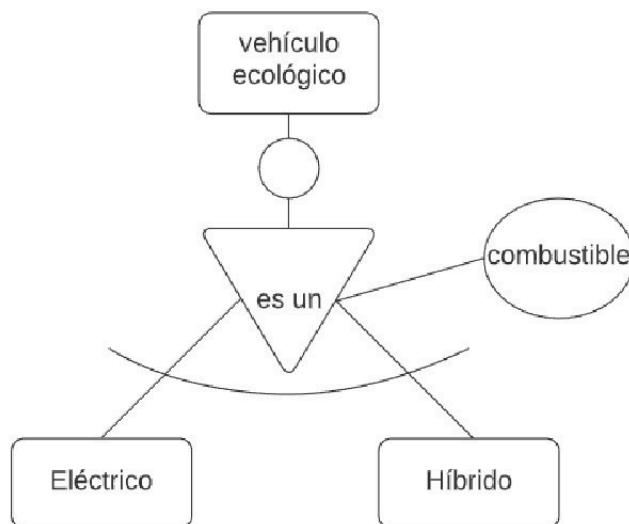
Relación que existe entre un tipo de entidad (supertipo) y los tipos de entidad más específicos (subtipos), en donde los atributos del supertipo son heredados por los subtipos.

La subdivisión es una necesidad bastante habitual en las BD. Esta relación especial se representa mediante un triángulo invertido con la base paralela al rectángulo que representa al supertipo.

Además, será necesario tener en cuenta y representar:

- El atributo del supertipo que no es heredado, será el que haga de discriminante, y se representa como un **atributo** de la generalización.
- Si los subtipos son de tipo exclusivo (pueden pertenecer a un subtipo u otro, pero no a los dos al la vez) se representan mediante un **arco**.
- Si el toda entidad del supertipo debe obligatoriamente ser de uno de los subtipos, esto se representa mediante un círculo o **elipse vacías**,

tal como se aprecia en la imagen:

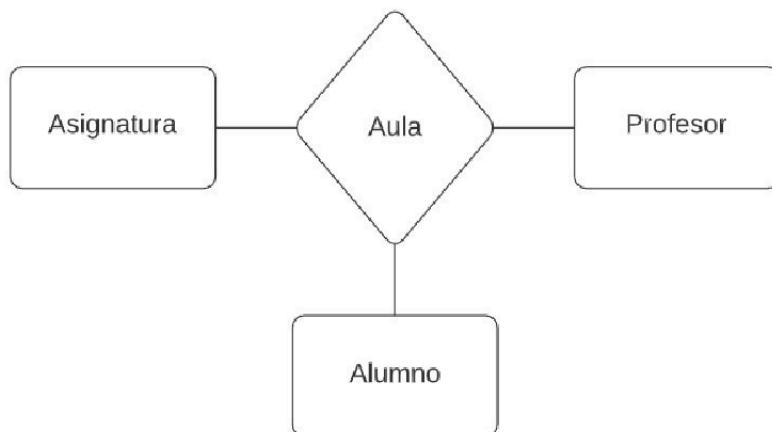


Siempre se verifica que toda ocurrencia de un subtipo es ocurrencia del supertipo: esto implica que las cardinalidades serán siempre (1,1) en el supertipo y (0,1) o (1,1) en el subtipo.

### 5.5. Relaciones ternarias,

O de orden superior: además de las relaciones habituales entre dos entidades (binarias) pueden existir otras que se consideran en el modelo E/R extendido.

Las relaciones ternarias se pueden descomponer en tres relaciones binarias. En general una relación n-aria se puede descomponer en n relaciones binarias, siendo esta última forma mucho más clara y fácil de interpretar en un diagrama E/R.



## 5.6. Construcción de un esquema E/R

Para elaborar un esquema E/R se pueden seguir las siguientes estrategias:

- **Ascendente:** Se parte de los atributos, y se van agrupando en entidades, y posteriormente se crean las relaciones.
- **Descendente:** En esta estrategia se refinan los conceptos de forma progresiva, concretizándolos.
- **Inside-Out:** Se parte de una vista del esquema, y se le van agregando elementos.
- **Mixto:** Es una combinación de las anteriores estrategias

## 6. Diseño lógico

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.

### 6.1. 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.

### 6.1.1. Transformación de entidades

Todas las entidades se pararán a tabla.

### 6.1.2. 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.

### 6.1.3. 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.

#### 6.1.4. 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.

## 7. 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.

### 7.1. 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.

## 8. 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.

### 8.1. A. 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.

### 8.2. B. 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:

### 8.3. C. 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:

$$R.X \rightarrow R.Y \text{ en vez de } R.X \rightarrow R.Z \rightarrow R.Y$$

#### 8.4. D. 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.

#### 8.5. E. 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.

#### 8.6. F. 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.*

## 9. Diseño físico

Una vez realizado el diseño lógico de la BD es necesario llevar a cabo su implementación en un DBMS concreto.

Es labor del administrador de la BD mejorar el rendimiento ajustando ciertos parámetros según las posibilidades que ofrezca el software concreto en el que se está implementando.

Además se debe realizar un estudio de la carga del sistema, especialmente de las consultas y los procesos de actualización que involucren múltiples registros.

Otras decisiones a considerar, serán los índices que se creen, las vistas, los formularios de introducción de datos y todas las posibles mejoras que se deban adoptar.

### 9.1. Buenas prácticas en el diseño físico

Según los autores Silberschatz, Korth y Sudarshan, en su libro fundamento de bases de datos (2014), para obtener un buen diseño, es necesario tener en cuenta adicionalmente las siguientes consideraciones:

- **Restricciones:** La aplicación de restricciones suele tener un coste computacional alto, es por ello que un buen diseño debe favorecer su cálculo.
- **Requisitos de autorización:** Un buen diseño debe facilitar la concesión de permisos.
- **Los flujos de datos y flujos de trabajo**
- **Obtener un buen rendimiento en las consultas**
- **Facilitar la posible evolución del sistema**

## 10. Conclusión

En este tema hemos visto el diseño de una base de datos siguiendo los pasos desde la creación del diagrama de E/R extendido, al diseño lógico normalizado.

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 implementarlo físicamente en un DBMS concreto. Para ello es conveniente hacer algunas medidas y provisiones sobre el funcionamiento de la BD.

### 10.1. 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)

#### Bachillerato:

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

## 11. 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)

