

## **EL MODELO DE DATOS**

La modelización consiste en representar el problema realizando múltiples abstracciones para asimilar toda la información de un problema, y de esta manera, generar una especie de mapa donde estén identificados todos los objetos de la base de datos.

Para modelar un problema de base de datos es necesario tener en cuenta las siguientes consideraciones:

- Casi con toda probabilidad, la persona que realiza la modelización es un analista informático, por lo que puede no ser un experto en el dominio del problema que debe resolver (Contabilidad, Medicina, Economía, etc.). Se ha de contar con la experiencia de un futuro usuario de la base de datos que conozca a fondo todos los pormenores del negocio, y que, a su vez, puede no tener conocimientos de informática.
- Hay que modelar siguiendo unas directrices o estándares, es decir, usando una filosofía estándar para que el resto de la comunidad informática pueda entender y comprender el modelo realizado. De esta manera, será posible aprovechar las herramientas informáticas software del mercado para realizar diseños.
- La base de datos estará gestionada por un SGBD que tendrá unas características técnicas, de esta manera, no se tratará igual la implantación de la base de datos en un sistema MySQL que en uno DB2.

Para satisfacer estas necesidades, se suele recurrir a tres modelados:

1. **El modelo conceptual.** Es un modelo que tiene un gran poder expresivo para poder comunicarse con un usuario que no es experto en informática. Tiene una gran potencia para representar el dominio del problema tal y como el usuario lo concibe. Uno de los modelos más utilizados es el modelo Entidad/Relación.
2. **El modelo lógico.** Este modelo es más técnico que el anterior. Los conceptos expresados por este modelo, suelen ser difíciles de entender por los usuarios y generalmente tienen traducción directa al modelo físico que entiende el SGBD. El modelo lógico elegido dependerá de la implementación de la base de datos, así, no es lo mismo modelizar una base de datos orientada a objetos, que modelizar una base de datos relacional. El modelo más se utiliza es el Modelo Relacional.

Instructor: **WILLINGTON RINCÓN HOYOS**

SENA Regional Quindío

Comercio y Turismo

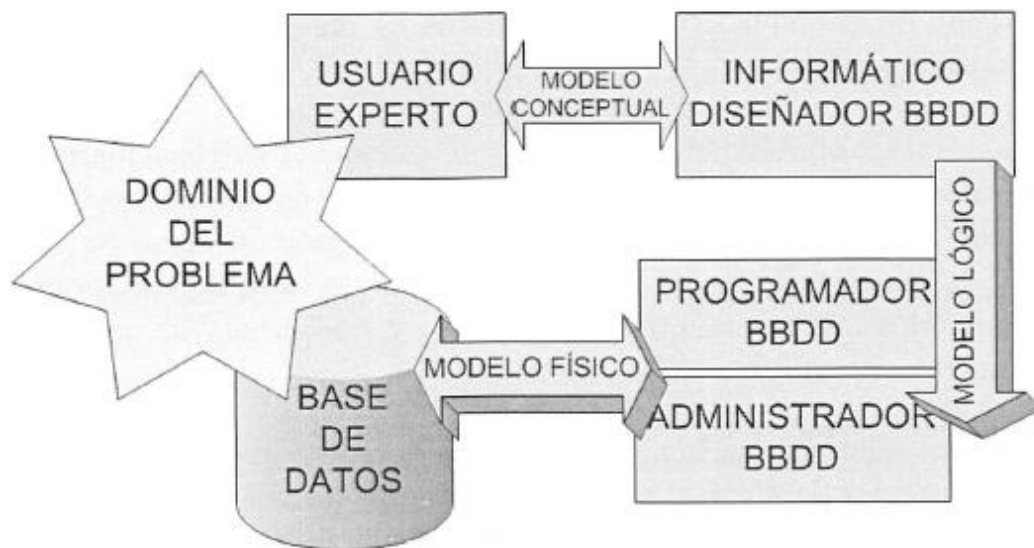
3. **El modelo físico.** Es el resultado de aplicar el modelo lógico a un SGBD concreto. Generalmente está expresado en un lenguaje de programación de BD tipo SQL. Se puede transformar el Modelo Relacional en un modelo físico a través del sublenguaje DDL de SQL.

La interacción entre estos tres modelos es fundamental para un diseño de calidad:

Primero, se negocia con el usuario el modelo conceptual.

Segundo, se pasa el modelo conceptual al modelo lógico, realizando una serie de transformaciones necesarias para adaptar el lenguaje del usuario al del gestor de base de datos.

Finalmente, se transforma el modelo lógico en físico, obteniendo de esta forma la base de datos final.



### Observación importante

En ocasiones, los diseñadores experimentados, realizan el diseño de la base de datos directamente en el modelo relacional. Esto puede representar un ahorro de tiempo si el problema a resolver es relativamente sencillo. Pero, generalmente, en problemas más complejos, saltarse el diseño conceptual y la opinión del usuario, da como resultado diseños incompletos e incoherentes.

## **Diagramas E/R**

Para representar el modelo conceptual se usará el modelo Entidad/Relación. Este modelo consiste en plasmar el resultado del análisis del problema mediante diagramas entidad-relación.

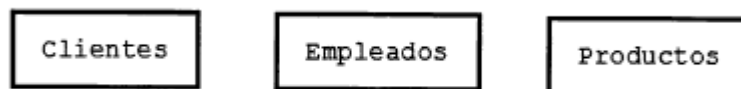
Estos diagramas fueron propuestos por Peter P. Chen a mediados de los años 70 para la representación conceptual de los datos y establecer qué relaciones existían entre ellos.

La notación se considera sencilla, y, precisamente, esta sencillez, permite representar el mundo real de forma que el usuario pueda validar si el modelo propuesto se ajusta perfectamente a la resolución del problema.

A continuación, se presentan las definiciones necesarias para comprender el modelo entidad relación.

### **Entidad**

Cualquier tipo de objeto o concepto sobre el que se recoge información; cosa, persona, concepto abstracto o suceso. Se representan mediante un cuadrado. Por ejemplo: coche, casa, empleado, cliente, etc. Las entidades se representan gráficamente mediante rectángulos y su nombre aparece en el interior (generalmente en singular). Un nombre de entidad solo puede aparecer una vez en el diagrama.



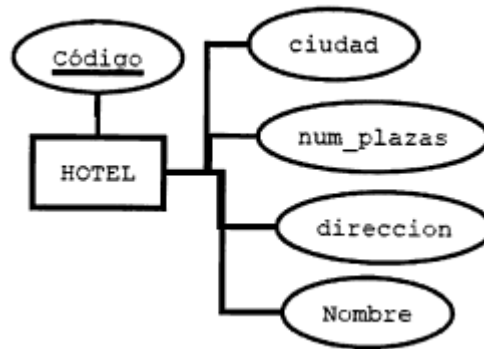
Hay dos tipos de entidades: fuertes (o regulares) y débiles. Las entidades débiles se representan mediante un cuadro doble. Una entidad débil es una entidad cuya existencia depende de la existencia de otra entidad. Una entidad fuerte es una entidad que no es débil, es decir, existe por méritos propios. Un ejemplo típico es la existencia de dos entidades para la representación de un pedido. Por un lado, la entidad pedido representa información genérica sobre el pedido como la fecha del pedido, fecha de envío, el estado, etc. Por otro lado, la entidad "Detalle de Pedido" recopila las líneas de información específica sobre los artículos y unidades pedidas. En este caso, "Detalle de Pedido" es una entidad débil, puesto que la eliminación del pedido implica la eliminación de las líneas de detalle asociadas al pedido, es decir, no tiene sentido almacenar información específica del pedido si se ha eliminado ese pedido.



### **Atributos y Dominios**

Los atributos de una entidad son las características o propiedades que la definen como entidad. Se representan mediante elipses conectadas directamente a la entidad.

Por ejemplo, para representar la entidad HOTEL, son necesarias sus características, esto es, el número de plazas disponibles, su dirección, la ciudad donde se encuentra, etc.



### **Atributo Clave**

En la figura anterior, aparece el atributo código, subrayado. Este atributo se denomina clave, y designa un campo que no puede repetir ninguna ocurrencia de entidad. Se dice, que este campo identifica unívocamente a una entidad, es decir, que con la sola referencia a un campo clave se tiene acceso al resto de atributos de forma directa. Ejemplo; El DNI es el campo clave de una persona, pues ninguna persona tiene el mismo DNI. Por tanto, si se especifica el DNI de esa persona se sabe exactamente a qué ocurrencia de persona se refiere. Todas las entidades fuertes deberían tener, al menos, un atributo clave. Nótese que una entidad puede formar la clave mediante varios atributos, en este caso, se dice que la clave de la entidad es la suma de esos atributos y que la entidad tiene una clave compuesta.

### **Dominios**

Cada una de las características que tiene una entidad pertenece a un dominio.

El dominio representa la naturaleza del dato, es decir, si es un número entero, una cadena de caracteres o un número real. Incluso naturalezas más complejas, como

Instructor: **WILLINGTON RINCÓN HOYOS**

SENA Regional Quindío

Comercio y Turismo

una fecha o una hora (con minutos y segundos). Por ejemplo, los siguientes atributos de la entidad empleado pertenecen a los siguientes dominios:

Atributo	Dominio
DNI	Cadena de Caracteres de longitud 10
Nombre	Cadena de Caracteres de longitud 50
Fecha_Nacimiento	Fecha
Dirección	Cadena de Caracteres de longitud 100
Sueldo	Números reales
Número de hijos	Números enteros
Departamento	Departamentos

Si un dominio se especifica mediante el tipo de datos, como en el caso de DNI, Nombre o Fecha\_Nacimiento se dice que se define por intensión. Si se especifica mediante un conjunto de valores, como en el dominio Departamentos, que puede tener los valores (RRHH, Informática, Administración o Contabilidad), la definición del dominio es por extensión.

### **Tipos de atributos**

Se pueden clasificar los atributos según las siguientes restricciones:

**Atributos obligatorios;** Un atributo debe tomar un valor obligatoriamente.

**Atributos opcionales:** Un atributo puede no tomar un valor porque sea desconocido en un momento determinado. En este caso, el atributo tiene un valor nulo.

**Atributos compuestos:** Un atributo compuesto es aquel que se puede descomponer en atributos más sencillos, por ejemplo, el atributo hora\_de\_salida se puede descomponer en dos (hora y minutos).

**Atributos univaluados:** Un atributo que toma un único valor.

**Atributos multivaluados:** Estos atributos pueden tomar varios valores, por ejemplo el atributo teléfono puede tomar los valores de un teléfono móvil y un teléfono fijo.

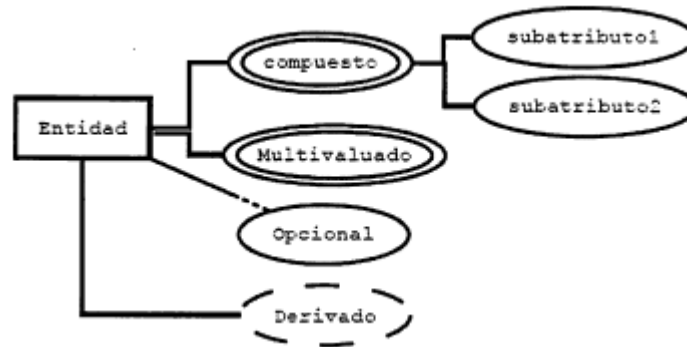
Instructor: **WILLINGTON RINCÓN HOYOS**

SENA Regional Quindío

Comercio y Turismo

**Atributo derivado:** Son aquellos cuyo valor se puede calcular a través de otros atributos. Por ejemplo, el atributo Edad, se puede calcular a partir de la fecha de nacimiento de una persona.

Al igual que con la mayoría de las notaciones, no existe unanimidad a la hora de dibujar en un diagrama los tipos de atributos. Una de las más extendidas entre los diseñadores de bases de datos es la siguiente:



### Otras notaciones para los atributos

Al igual que para las entidades, los atributos tienen multitud de notaciones, y, aunque la original adoptada por Peter Chen es la más usada hasta ahora, por simplificar la construcción de mapas a través de herramientas software, se opta por usar otras notaciones que producen mapas más manejables. Por ejemplo, la herramienta MySQL Workbench utiliza una sintaxis muy similar a la que usa la notación UML para representar las características de un objeto:

