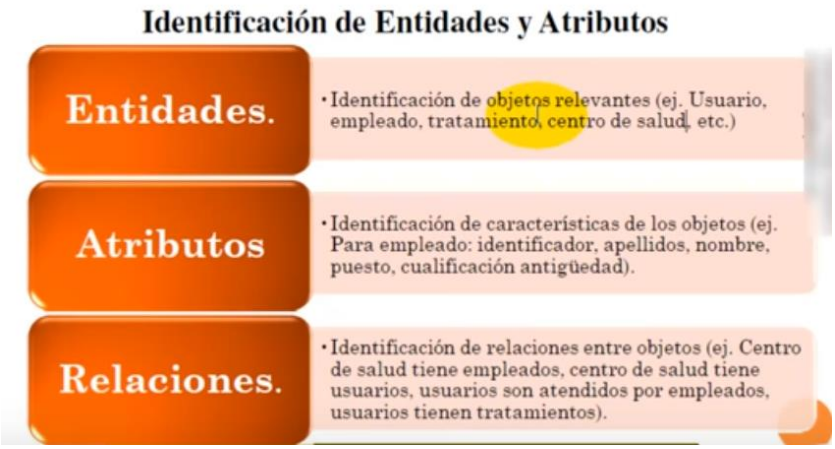
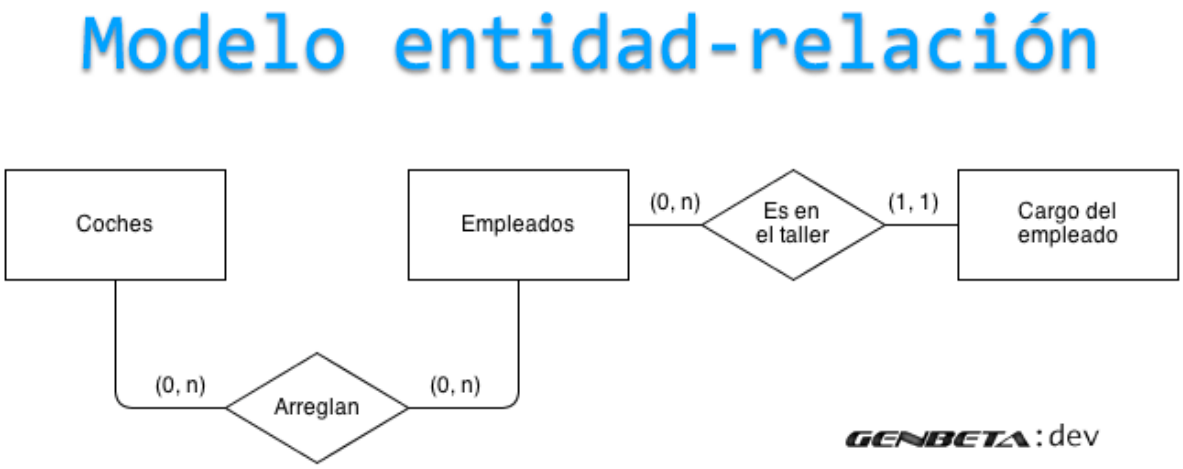
RELACION UNO A MUCHOS ;MUCHOS A UNO; MUCHOS A MUCHOS





BASE DE DATOS

Las **bases de datos** son un gran pilar de la programación actual, ya que nos permiten almacenar y usar de forma rápida y eficiente cantidades ingentes de datos con cierta facilidad. En la actualidad se usa de forma mayoritaria las **bases de datos relacionales** (dominadas por distintos gestores a través del lenguaje **SQL**, en gran medida).

Pero ahora vamos a dar un pequeño repaso a lo más esencial del **modelo entidad-relación**, que es y ha sido durante años la mejor forma de representar la estructura de estas bases de datos relacionales (o de representar sus esquemas).

¿Qué es el modelo entidad-relación?

Como ya he comentado este modelo es solo y exclusivamente un método del que disponemos para diseñar estos esquemas que posteriormente debemos de implementar en un gestor de *BBDD* (bases de datos). Este modelo se representa a través de diagramas y está formado por varios elementos.

Este modelo habitualmente, además de disponer de un diagrama que ayuda a entender los datos y como se relacionan entre ellos, debe de ser completado con un pequeño resumen con la lista de los atributos y las relaciones de cada elemento.

Elementos del modelo entidad-relación

Entidad

Las entidades representan *cosas* u *objetos* (ya sean reales o abstractos), que se diferencian claramente entre sí.

Para poder seguir un ejemplo durante el artículo añadiré ejemplos sobre un taller mecánico, donde se podría crear las siguientes entidades:

- **Coches** (objeto físico): contiene la información de cada taller.
- **Empleado** (*objeto* físico): información de los trabajadores.
- **Cargo del empleado** (*cosa* abstracta): información de la función del empleado.

Estas entidades se representan en un diagrama con un rectángulos, como los siguientes.

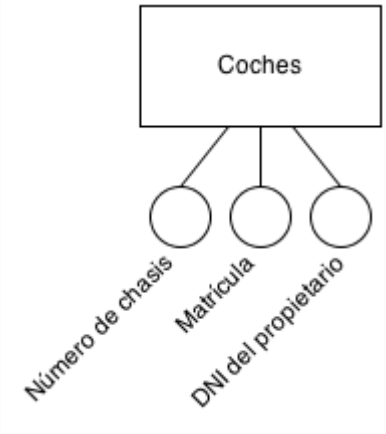
Atributos

Los atributos definen o identifican las características de entidad (**es el contenido de esta entidad**). Cada entidad contiene distintos atributos, que dan información sobre esta entidad. Estos atributos pueden ser de distintos tipos (numéricos, texto, fecha...).

Siguiendo el ejemplo de antes podemos analizar los atributos de nuestra entidad "**Coches**", que nos darán información sobre los coches de nuestro supuesto taller.

Unos posibles atributos serían los siguientes: *número de chasis*, *matrícula*, *DNI del propietario*, *marca*, *modelo* y muchos otros que complementen la información de cada coche.

Los atributos se representan como círculos que descienden de una entidad, y no es necesario representarlos todos, sino los más significativos, como a continuación.



En un modelo relacional (ya implementado en una base de datos) una ejemplo de tabla dentro de una *BBDD* podría ser el siguiente.

Número de chasis	**Matrícula**	**DNI del propietario**
5tfem5f10ax007210	4817 BFK	45338600L
6hsen2j98as001982	8810 CLM	02405068K
5rgsb7a19js001982	0019 GGL	40588860J

Este ejemplo es con tres atributos, pero un coche podría tener cientos (si fuese necesario) y seguirían la misma estructura de columnas, tras implementarlo en una *BBDD*.

Relación

Es un vínculo que nos permite definir una dependencia entre varias entidades, es decir, nos permite exigir que varias entidades compartan ciertos atributos de forma indispensable.

Por ejemplo, los empleados del taller (de la entidad "**Empleados**") tienen un cargo (según la entidad "**Cargo del empleado**"). Es decir, un atributo de la entidad "*Empleados*" especificará que cargo tiene en el taller, y tiene que ser idéntico al que ya existe en la entidad "*Cargo del empleado*".

Las relaciones se muestran en los diagramas como rombos, que se unen a las entidades mediante líneas.

Yo, bajo mi punto de vista, entiendo mejor esto en una tabla (de una implementación en una *BBDD*), por lo que voy a poner el ejemplo de como se representaría (resaltada la relación, que posteriormente veremos como se haría).

Empleados

Nombre	DNI	Cargo
Carlos Sánchez	45338600L	**001**
Pepe Sánchez	02405068K	**002**
Juan Sánchez	40588860J	**002**

Cargo del empleado

ID del cargo	Descripción
001	Jefe de taller
002	Mecánico

Relaciones de cardinalidad

Podemos encontrar distintos tipos de relaciones según como participen en ellas las entidades. Es decir, en el caso anterior cada empleado puede tener un cargo, pero un mismo cargo lo pueden compartir varios empleados.

Esto complementa a las representaciones de las relaciones, mediante un intervalo en cada extremo de la relación que especifica cuantos *objetos* o *cosas* (de cada entidad) pueden intervenir en esa relación.

Uno a uno: Una entidad se relaciona únicamente con otra y viceversa. Por ejemplo, si tuviésemos una entidad con distintos chasis y otra con matrículas deberíamos de determinar que cada chasis solo puede tener una matrícula (y cada matrícula un chasis, ni más en ningún caso).

Uno a varios o **varios a uno**: determina que un registro de una entidad puede estar relacionado con varios de otra entidad, pero en esta entidad existir solo una vez. Como ha sido en el caso anterior del trabajador del taller.

Varios a varios: determina que una entidad puede relacionarse con otra con ninguno o varios registros y viceversa. Por ejemplo, en el taller un coche puede ser reparado por varios mecánicos distintos y esos mecánicos pueden reparar varios coches distintos.

Los indicadores numéricos indican el primero el número mínimo de registros en una relación y posteriormente el máximo (si no hay límite se representa con una "*n*").

Claves

Es el atributo de una entidad, al que le aplicamos una restricción que lo distingue de los demás registros (no permitiendo que el atributo específico se repita en la entidad) o le aplica un vínculo (exactamente como comentábamos en las relaciones). Estos son los distintos tipos:

Superclave: aplica una clave o restricción a varios atributos de la entidad, para así asegurarse que en su conjunto no se repitan varias veces y así no poder entrar en dudas al querer identificar un registro.

Clave primaria: identifica inequívocamente un solo atributo no permitiendo que se repita en la misma entidad. Como sería la matrícula o el número de chasis de un coche (no puede existir dos veces el mismo).

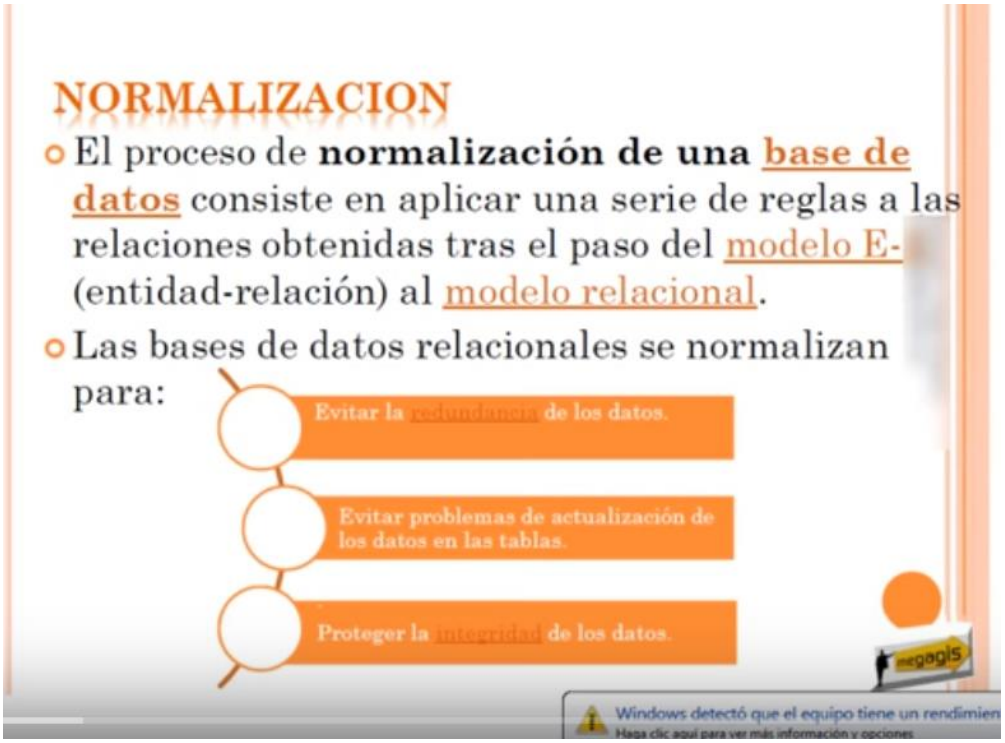
Clave externa o **clave foránea**: este campo tiene que estar estrictamente relacionado con la clave primaria de otra entidad, para así exigir que exista previamente ese clave. Anteriormente hemos hablado de ello cuando comentábamos que un empleado indispensablemente tiene que tener un cargo (que lo hemos representado numéricamente), por lo cual si intentásemos darle un cargo inexistente el gestor de bases de datos nos devolvería un error.

Resumen

Esto ha sido solo un repaso por encima de lo que es el **modelo entidad-relación**, sin entrar en grandes detalles.

También, bajo mi punto de vista, creo que es una buena forma de diseñar correctamente las bases de datos, aunque algunas veces resulta más rápido implementarlo directamente en nuestro gestor de *BBDD* sin la necesidad de crear un gran diagrama, sino usando notas más simples.

NORMALIZACION DE UNA BASE DE DATOS



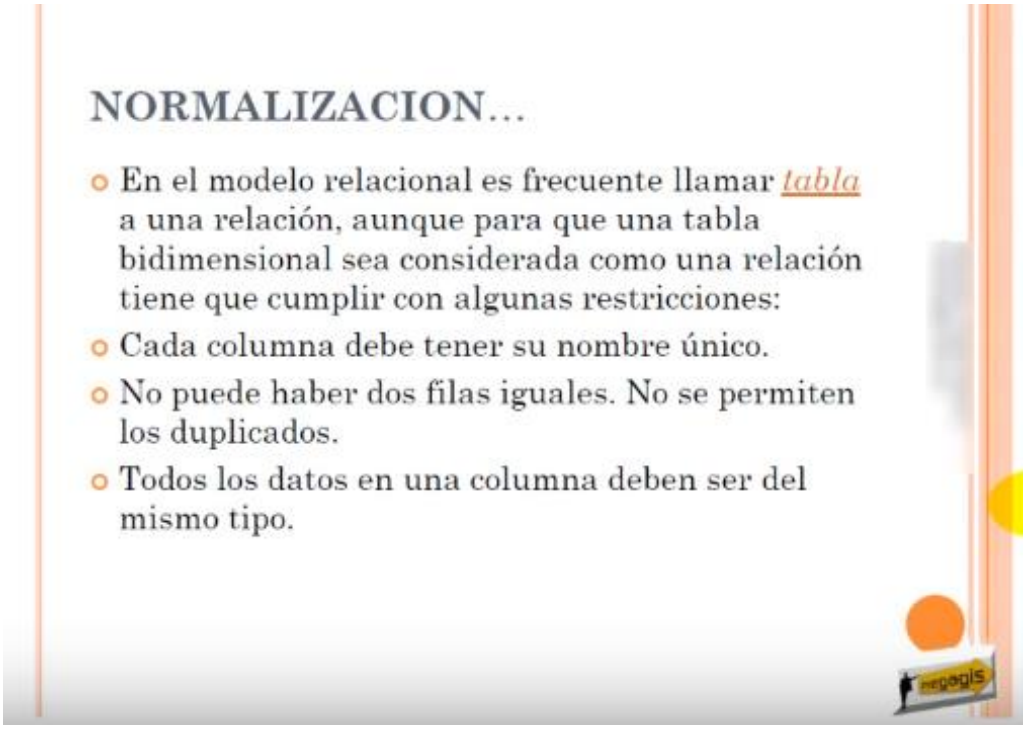


TABLA QUE NO CUMPLE

CodLibro	Titulo	Autor	Editorial	NombreLector	FechaDev
1001	Variable compleja	Murray Spiegel	McGraw Hill	Pérez Gómez, Juan	15/04/2005
1004	Visual Basic 5	E. Petroustsos	Anaya	Rios Terán, Ana	17/04
1005	Estadistica	Murray Spiegel	McGraw Hill	Roca, René	16/04
1006	Oracle University	Nancy Greenberg y Priya Nathan	Oracle Corp.	García Roque, Luis	20/04
1007	Clipper 5.01	Ramalho	McGraw Hill	Pérez Gómez, Juan	18/04/2005

1NF

- Esta tabla no cumple el requisito de la Primera Forma Normal (1NF) de sólo tener campos atómicos, pues el nombre del lector es un campo que puede (y conviene) descomponerse en apellido paterno, apellido materno y nombres. Tal como se muestra en la siguiente tabla.

CodLibro	Titulo	Autor	Editorial	Paterno	Materno	Nombres	FechaDev
1001	Variable compleja	Murray Spiegel	McGraw Hill	Pérez	Gómez	Juan	15/04/2010
1004	Visual Basic 5	E. Petroustsos	Anaya	Ríos	Terán	Ana	17/04/2010
1005	Estadística	Murray Spiegel	McGraw Hill	Roca		René	16/04/2010
1006	Oracle University	Nancy Greenberg	Oracle Corp.	García	Roque	Luis	20/04/2010
1006	Oracle University	Priya Nathan	Oracle Corp.	García	Roque	Luis	20/04/2010
1007	Clipper 5.01	Ramalho	McGraw Hill	Pérez	Gómez	Juan	18/04/2010

2NF

- La Segunda Forma Normal (2NF) pide que no existan dependencias parciales o dicho de otra manera, todos los atributos no clave deben depender por completo de la clave primaria. Actualmente en nuestra tabla tenemos varias dependencias parciales si consideramos como atributo clave el código del libro.
- Por ejemplo, el título es completamente identificado por el código del libro, pero el nombre del lector en realidad no tiene dependencia de este código, por tanto estos datos deben ser trasladados a otra tabla.

CodLibro	Titulo	Autor	Editorial
1001	Variable compleja	Murray Spiegel	McGraw Hill
1004	Visual Basic 5	E. Petroustsos	Anaya
1005	Estadística	Murray Spiegel	McGraw Hill
1006	Oracle University	Nancy Greenberg	Oracle Corp.
1006	Oracle University	Priya Nathan	Oracle Corp.
1007	Clipper 5.01	Ramalho	McGraw

La nueva tabla sólo contendrá datos del lector.

CodLector	Paterno	Materno	Nombres
501	Pérez	Gómez	Juan
502	Ríos	Terán	Ana
503	Roca		René
504	García	Roque	Luis

- Hemos creado una tabla para contener los datos del lector y también tuvimos que crear la columna CodLector para identificar unívocamente a cada uno. Sin embargo, esta nueva disposición de la base de datos necesita que exista otra tabla para mantener la información de qué libros están prestados a qué lectores. Esta tabla se muestra a continuación:

CodLibro	CodLector	FechaDev
1001	501	15/04/2005
1004	502	17/04/2005
1005	503	16/04/2005
1006	504	20/04/2005
1007	501	18/04/2005

3FN

- Para la Tercera Forma Normal (3NF) la relación debe estar en 2NF y además los atributos no clave deben ser mutuamente independientes y dependientes por completo de la clave primaria. También recordemos que dijimos que esto significa que las columnas en la tabla deben contener solamente información sobre la entidad definida por la clave primaria y, por tanto, las columnas en la tabla deben contener datos acerca de una sola cosa.
- En nuestro ejemplo en 2NF, la primera tabla conserva información acerca del libro, los autores y editoriales, por lo que debemos crear nuevas tablas para satisfacer los requisitos de 3NF.

CodLibro	Titulo
1001	Variable compleja
1004	Visual Basic 5
1005	Estadistica
1006	Oracle University
1007	Clipper 5.01

CodAutor	Autor
801	Murray Spiegel
802	E. Petroustsos
803	Nancy Greenberg
804	Priya Nathan
806	Ramalho

CodEditorial	Editorial
901	McGraw Hill
902	Anaya
903	Oracle Corp.

- **Aunque hemos creado nuevas tablas para que cada una tenga sólo información acerca de una entidad, también hemos perdido la información acerca de qué autor ha escrito qué libro y las editoriales correspondientes, por lo que debemos crear otras tablas que relacionen cada libro con sus autores y editoriales.**

CodLibro	codAutor
1001	801
1004	802
1005	801
1006	803
1006	804
1007	806

CodLibro	codEditorial
1001	901
1004	902
1005	901
1006	903
1007	901

CodLibro	CodLector	FechaDev
1001	501	15/04/2010
1004	502	17/04/2010
1005	503	16/04/2010
1006	504	20/04/2010
1007	501	18/04/2010

CodLector	Paterno	Materno	Nombres
501	Pérez	Gómez	Juan
502	Ríos	Terán	Ana
503	Roca		René
504	García	Roque	Luis

Practica.

Ejemplo de Normalización

Base de Datos sin Normalizar

Clínica I

Clínica I

Nombre Paciente	Dirección	Teléfono	Nombre Profesional	Fecha
A1	C/Tupilan	9999	P1	25/12/2010
A1	C/Tupilan	9999	P1	28/12/2010
A1	C/Tupilan	9999	P2	1/1/2010
A2	C/Amalopa	8888	P1	25/12/2010

Consultas:

- Queremos saber la dirección y el teléfono de todos los pacientes que hayan sido tratados por el especialista P1.
- Nombre, dirección de todos los pacientes que tienen consulta el 25/12/2010.
- …

Bases de Datos Relacionales.

Ejemplo de Normalización

Base de Datos Normalizada

Clínica I

Pacientes

DNI	Apellidos	Nombre	Dirección	Teléfono
12345678-s	García Romeral	Benito	C/tupilán	9999
98765432-d	Robledo Tuk	Gloria	C/Amalopa	8888

Profesionale

DNI	Apellidos	Nombre	Teléfono	Dirección
78732733-Q	Humero Yodal	Lorenzo	78465	C/Penutia
464575852-M	Nadal Bingo	Marta	63633	C/Gisarol

Consulta

DNI_paciente	DNI_profesional	Fecha
12345678-s	78732433-Q	25/12/2010
12345678-s	78732433-Q	31/12/2010
98765432-d	464575852-M	25/12/2010

Consultas:

Se pueden hacer las mismas consultas.