

Diseño de Base de Datos Relacionales

IES EL PALO

2º GRADO SUPERIOR IMAGEN PERSONAL

Índice

Introducción	3
1.- Diseño conceptual.....	4
1.1- Recogida y análisis de datos.....	4
1.2- Identificación de las entidades (tablas).....	4
1.3- Identificación de los tipos de relaciones (Cardinalidad)	5
1.4- Identificación de atributos (campos)	7
1.5- Identificación de las claves principales	7
1.6- Identificación de entidades fuertes y débiles	7
1.7- Identificación Generalización y Especialización (si las hay).....	9
2.- Diseño lógico	10
2.1- Diagrama E-R.....	10
2.2- Normalización	11
3.- Diseño físico	13
3.1- Verificación	13
3.2- Implementación	13

Introducción

Una base de datos o banco de datos (BD) es un conjunto de datos pertenecientes a un mismo sistema de información y almacenados de forma organizada para su posterior uso.

El diseño de bases de datos es el proceso por el que se determina la organización de una base de datos, incluidos su estructura, contenido y las aplicaciones que se han de desarrollar.

Ventajas

- Evitar la **duplicidad de registros o redundancia**
- **Garantiza la integridad referencial**, así, al eliminar un registro elimina todos los registros relacionados dependientes
- **Flexibilidad**
- **Accesibilidad**
- Favorece la **normalización** por ser más comprensible y aplicable
- **Seguridad de datos**

Así, el diseño de una Base de Datos se descompone en:

1. **Diseño conceptual.**
2. **Diseño lógico.**
3. **Diseño físico.**

1.- Diseño conceptual

Es una descripción del mundo real donde se hace recogida y análisis de requisitos, independientemente del SGBD (*motor de base de datos*) que se vaya a utilizar para manipularla.

1.1- Recogida y análisis de datos

Es pensar en el mundo real que queremos representar.

No hay un procedimiento claro y universal, aunque sí se pueden dar algunas directrices generales.

El proceso de análisis consiste esencialmente en obtener una solución a un problema de diseño planteado a partir de las:

- **Especificaciones**
- **Requisitos**
- **Necesidades**

Hablar con el cliente e intentar dejar claros los parámetros y objetivos del problema o proceso a modelar. Por supuesto, tomar buena nota de todo.

- **Entrevistas con los usuarios de todos los niveles**
- **Análisis de los documentos a generar**
- **Objetivos de la organización.**
- **Estudio de las reglas de la empresa.**
- **Etc...**

En resumen, tenemos que recopilar todas las fuentes de datos de la empresa.

El modelo lo componen los siguientes elementos:

- 1) Identificación de entidades.
- 2) Identificación de relaciones indicando grado.
- 3) Identificación de atributos de entidades.
- 4) Identificación de claves principales y ajenas de entidades.
- 5) Identificación de entidades fuertes y débiles.
- 6) Identificación de generalización y especialización.

1.2- Identificación de las entidades (tablas)

Una entidad es cualquier objeto del mundo real con existencia física o conceptual propia, que es capaz de ser descripto y del cual se quiere tener información.

¿Cómo se identifica las entidades?

Brevemente consiste en los siguientes pasos:

- ✓ Se hace una lista de los sustantivos y verbos que aparecen.
- ✓ Los sustantivos son posibles entidades o atributos.
- ✓ Los verbos son posibles relaciones.
- ✓ Analizando las frases se determina la cardinalidad de las relaciones y otros detalles.

Las entidades se representan con un rectángulo



Ejemplo1: A partir del siguiente enunciado se desea realiza el modelo entidad-relación.

“Una empresa vende productos a varios clientes. Se necesita conocer los datos personales de los clientes (nombre, apellidos, dni, dirección y fecha de nacimiento).

Cada producto tiene un nombre y un código, así como un precio unitario. Un cliente puede comprar varios productos a la empresa, y un mismo producto puede ser comprado por varios clientes.

Los productos son suministrados por diferentes proveedores. Se debe tener en cuenta que un producto sólo puede ser suministrado por un proveedor, y que un proveedor puede suministrar diferentes productos. De cada proveedor se desea conocer el NIF, nombre y dirección”.

Ejemplo2: A partir del siguiente enunciado diseñar el modelo entidad-relación.

“Se desea diseñar la base de datos de un Instituto. En la base de datos se desea guardar los datos de los profesores del Instituto (DNI, nombre, dirección y teléfono).

Los profesores imparten módulos, y cada módulo tiene un código y un nombre. Cada alumno está matriculado en uno o varios módulos.

De cada alumno se desea guardar el nº de expediente, nombre, apellidos y fecha de nacimiento. Los profesores pueden impartir varios módulos, pero un módulo sólo puede ser impartido por un profesor.

Cada módulo tiene un grupo de alumnos, uno de los cuales es el delegado del grupo”

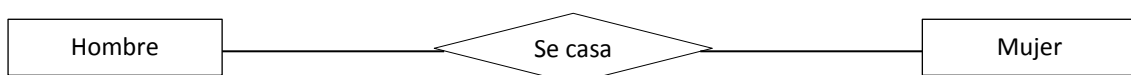
1.3- Identificación de los tipos de relaciones (Cardinalidad)

La cardinalidad es un tipo de restricción que se utiliza para comprobar y mantener la calidad de los datos

Las relaciones pueden tener los siguientes tipos de cardinalidad:

Las relaciones se representan con **rombo**

- ✓ Uno a uno 1:1



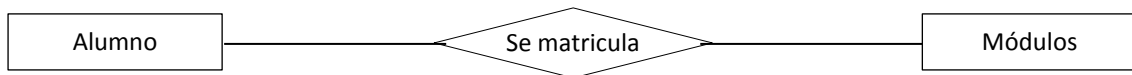
NOTA: Cuando la relación es de 1:1, la entidad con menos atributos (o campos) traslada todos sus atributos en la otra y desaparece.

- ✓ Uno a muchos 1:N



NOTA: Cuando la relación es de 1:N, la entidad del lado 1 presta su clave o sus claves principales (en el caso que está compuesta, es decir 2 o más atributos) a la otra entidad como clave ajena.

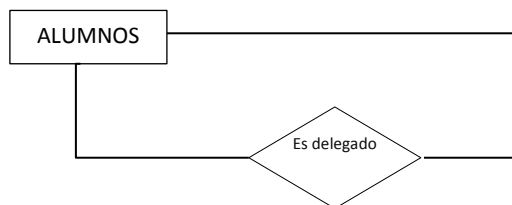
- ✓ Muchos a muchos N:N



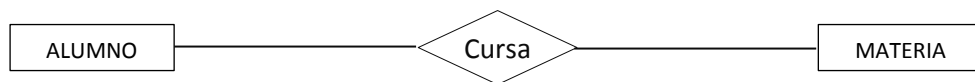
NOTA: Cuando la relación es de N:N, la relación (EL ROMBO) se convierte en entidad entre ellas, heredando las claves principales de ambos como claves principales y ajena.

Grado: número de conjuntos de entidades que intervienen en una interrelación.

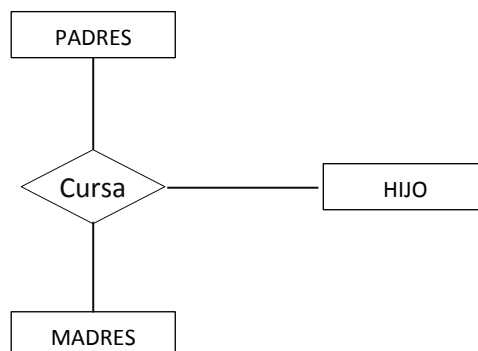
- Unaria o reflexiva



- Binarias



- Ternarias



1.4- Identificación de atributos (campos)

Los **atributos** son las características que definen o identifican a una entidad. Estas pueden ser muchas, y el diseñador solo utiliza o implementa las que considere más relevantes.

Lo más sencillo es preguntarse, para cada entidad y cada relación, ¿qué información se quiere saber de...?

Los atributos se representan con **elipses**



Ej1: El DNI, el apellido y el nombre de una persona.

Ej2: El código, la descripción y el precio de un producto.

Atributo multivaluado: (o multivalorado) se dice del atributo tal que para una misma entidad puede tomar varios valores diferentes, es decir, varios valores del mismo dominio.

Los atributos multivaluado se representan con doble rombos



1.5- Identificación de las claves principales

La **clave principal:** es un o varios atributos que identifican de forma unívoca una entidad.

En este paso, se trata de encontrar todos los identificadores de cada una de las entidades.

Los identificadores pueden ser **simples** (un solo atributo) o **compuestos** (varios atributos).

- Este clave nunca se repite.
- Nunca debe ser nulo.

Los campos claves serán representados gráficamente a través de un **asterisco (*)** o una llave a la izquierda de los mismos.

Ej: El DNI podría ser la clave principal en una tabla de empleados, ya que identifica a un empleado de otro.

Nota: Una clave foránea es una referencia a una clave en otra tabla, determina la relación existente en dos tablas. Las claves foráneas no necesitan ser claves únicas en la tabla donde están y sí a donde están referenciadas.

1.6- Identificación de entidades fuertes y débiles

Cuando una entidad participa en una relación puede adquirir un papel fuerte o débil.

- ✓ **Entidad fuerte** (también conocida como entidad regular)

Es aquella que sí puede ser identificada unívocamente. En los casos en que se requiera, se puede dar que una entidad fuerte "preste" algunos de sus atributos a una entidad débil para que esta última se pueda identificar.

Se representan mediante un doble rectángulo; es decir, un rectángulo con doble línea.



✓ **Entidad débil**

Una entidad débil es aquella que no puede existir sin participar en la relación; es decir, aquella que no puede ser unívocamente identificada solamente por sus atributos.

Se representan mediante un doble rectángulo; es decir, un rectángulo con doble línea.

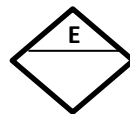


Se puede hablar de la existencia de 2 tipos de dependencias en las entidades débiles:

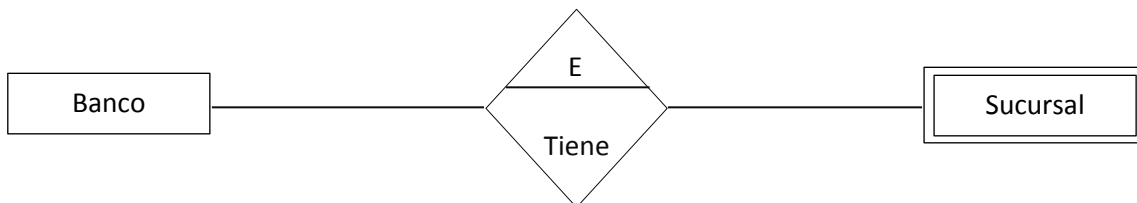
✓ **Dependencia por existencia**

Cuando las ocurrencias de una entidad débil, no pueden existir si desaparece la ocurrencia de la entidad fuerte de la cual dependen.

El rombo de la relación va etiquetado con **E** de existencia



Ejemplo:

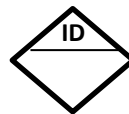


✓ **Dependencia por identidad**

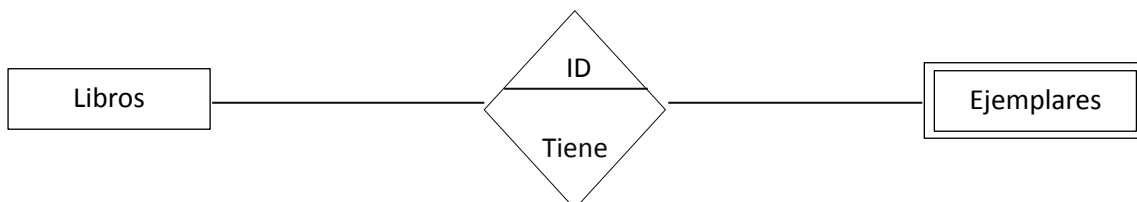
Cuando además de cumplirse la condición anterior, la entidad débil no se pueden identificar únicamente mediante los atributos propios de la misma y debe añadir la clave de la entidad fuerte de la cual dependen.

Una dependencia en *identificación* es siempre una dependencia en *existencia* (no ocurre el contrario)

El rombo de la relación va etiquetado con **ID** de identificación



Ejemplo:



1.7- Identificación Generalización y Especialización (si las hay)

Se da cuando se tienen varias entidades con características comunes y pueden crearse una entidad superior que tenga la información general de la aplicación. En otras palabras la generalización es el proceso inverso de la especialización.

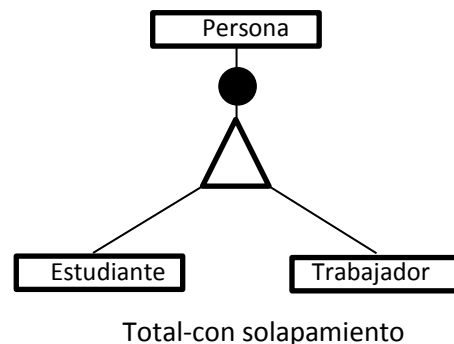
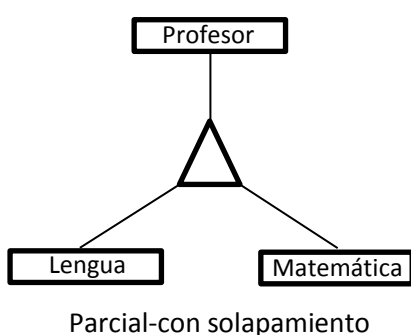
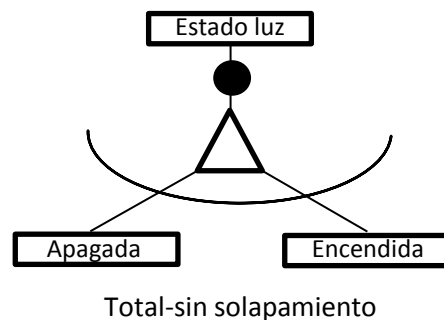
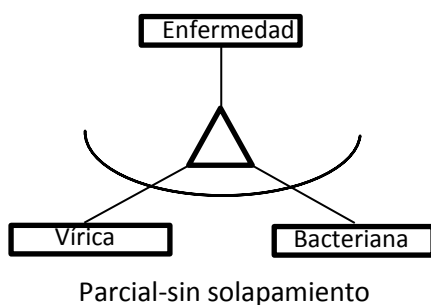
Cada subentidad puede poseer atributos propios.

Los subentidades heredan los atributos que pudiera tener la entidad general.

Las generalizaciones/Especializaciones pueden ser:

- ✓ **Total**: Cuando cada instancia o elemento concreto debe ser obligatoriamente de alguno de los subentidades especificados.
- ✓ **Parcial**: Cuando existe la posibilidad de otra subentidad que no es ninguno de los hemos especificados
- ✓ **Inclusiva** (con solapamiento): Si una instancia o elemento concreto de la entidad puede ser de más de un subentidad a la vez.
- ✓ **Exclusiva** (sin solapamiento): Si una instancia o elemento concreto de la entidad solo puede ser de uno de los subentidades.

Solo se hace cuando hay la posibilidad de dividir en subentidad o subtipo la entidad padre.



Una relación de especialización parcial puede fácilmente convertirse en total añadiendo un nuevo subtipo "otros".

2.- Diseño lógico

El diseño lógico es el proceso de construir un esquema de la información que utiliza la empresa, basándose en un modelo conceptual de base de datos específico

Podemos decir que es la conversión del esquema conceptual de datos en un esquema lógico.

Transformamos las entidades y relaciones obtenidas del modelo anterior en tablas.

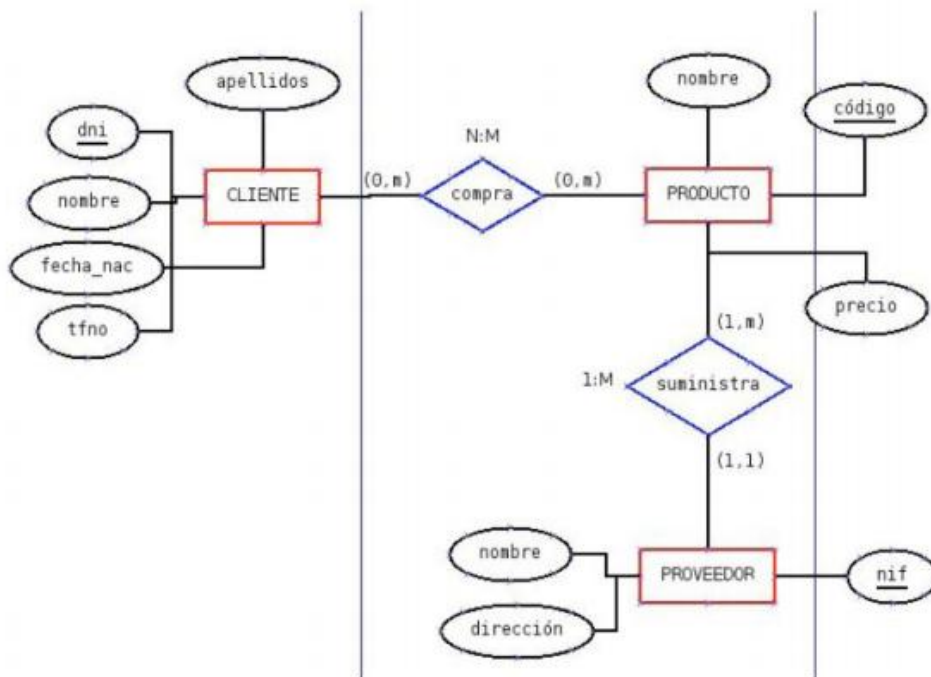
Dominio: Rango o conjunto de posibles valores de un atributo y se representan con hexágonos
En cada atributos.



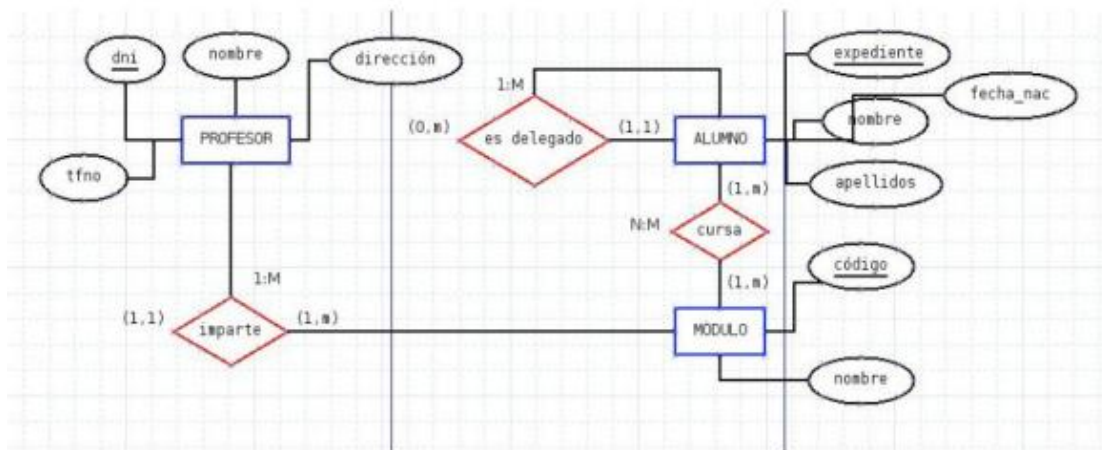
Utilizando la simbología vista, representaremos gráficamente todas las entidades, atributos relevantes, claves principales, y relaciones, indicando cardinalidad y grado.

2.1- Diagrama E-R

Enunciado 1:



Enunciado 1:



NB: OLVIDAROS DE ESTA PARTE DE LA NORMALIZACIÓN

2.2- Normalización

El proceso de normalización de bases de datos consiste en aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo entidad-relación al modelo relacional.

Las bases de datos relacionales se normalizan para:

- Evitar la redundancia de los datos.
- Evitar problemas de actualización de los datos en las tablas.
- Proteger la integridad de los datos.

Objetivos de la normalización de base de datos

Al proceder a la normalización de base de datos hay que plantearse 4 metas:

- Organizar los datos en grupos lógicos, de tal manera que cada grupo describa una pequeña parte del todo.
- Minimizar la cantidad de datos duplicados almacenados en una base de datos.
- Perfeccionar la organización de los datos de tal manera que, cuando se necesite introducir modificaciones, el cambio sólo deba aplicarse en un lugar.
- Construir una base de datos a la que se pueda acceder de forma rápida y donde sea posible manipular los datos con la máxima eficiencia y sin comprometer su integridad.

Nota: Cuando no existe normalización, se presentan anomalías en la base de datos. Estos problemas que ocasionan problemas al momento de insertar, modificar o eliminar datos.

✓ **Primera forma normal**

La primera forma normal (1FN), requiere que los datos sean atómicos. En otras palabras, la 1FN prohíbe a un campo contener más de un dato de su dominio de columna. También exige que todas las tablas deban tener una clave primaria.

✓ Segunda forma normal

Una tabla está en 2NF, si lo está ya en 1FN y todos sus atributos de una relación dependen funcionalmente de las claves candidatas de la relación (y no de partes de ellas).

Nota: si clave primaria no está compuesta, está ya en 2FN

✓ Tercera forma normal

La tercera forma normal (3NF) es una forma normal usada en la normalización de bases de datos. Para que la tabla esté en 3NF, se debe cumplir que la tabla ya esté en 2NF. Adicionalmente, ningún atributo no primario de la tabla debe ser dependiente transitivamente de una clave primaria.

Existen más formas normales: Forma normal Óbice Boyce-Codd (BCNF), (4FN), (5FN),.... Pero algunos autores opinan que a partir de todas estas formas normales se puede producir pérdida de dependencias.

En Resumen

- 1NF significa que tus relaciones tienen un número de atributos fijos y atómicos.
- 2NF significa que los atributos dependen de toda la clave primaria y no de parte de ella.
- 3NF significa que los atributos dependen directamente de la clave primaria (y no indirectamente a través de otro atributo) Estos tres puntos es todo lo que os hace falta recordar

3.- Diseño físico

Es el proceso de producir la descripción de la implementación de la base de datos en el ordenador.

Para llevar a cabo esta etapa, se debe haber decidido cuál es el **SGBD** que se va a utilizar, ya que el esquema físico se adapta a él.

En general, el propósito del diseño físico es describir cómo se va a implementar físicamente el esquema lógico obtenido en la fase anterior. Concretamente, en el modelo relacional, esto consiste en:

- Obtener un conjunto de relaciones (tablas) y las restricciones que se deben cumplir sobre ellas.
- Determinar las estructuras de almacenamiento y los métodos de acceso que se van a utilizar para conseguir unas prestaciones óptimas.
- Diseñar el modelo de seguridad del sistema.

Los nombres de los campos deben ser preferentemente nombres cortos y representativos de lo que almacenan, sin usar caracteres especiales, acentos, y espacios en blanco.

RECOMENDADO: guion bajo y en minúscula

3.1- Verificación

Aunque las verificaciones deben hacerse en todos y cada uno de los pasos. Al final revisaremos de nuevo los resultados obtenidos a fin de identificar si existieran relaciones redundantes y comprobar si el modelo conceptual obtenido satisface todas las necesidades de información del sistema en cuestión.

Se dice que una relación es redundante cuando su eliminación no implica pérdida semántica porque existe la posibilidad de realizar la misma asociación por medio de otras relaciones. Para que una relación sea redundante, es condición necesaria, aunque no suficiente, que forme parte de un ciclo. Por lo que habrá que estudiar con detenimiento los ciclos.

3.2- Implementación

Enunciado 1:

CLIENTES (dni, nombre, apellidos, fecha_nacimiento, telefono)

PRODUCTOS (código, nombre, precio, cif_proveedor)

PROVEEDORES (cif, nombre, dirección,...)

COMPRAS (dni_cliente, cod_producto)

Enunciado 2:

PROFESORES (dni, nombre, dirección, telefono)

MODULOS (código, nombre, dni_profesor)

ALUMNOS (expediente, nombre, apellidos, fecha_nacimiento, expediente_delegado)