

MÓDULO 4.1. MODELO ENTIDAD RELACIÓN



MINISTERIO
DE EDUCACIÓN
Y FORMACIÓN PROFESIONAL



Índice:

1. Introducción
2. Fases del diseño
3. Modelo E-R
4. Formas normales

1

Introducción

1. Introducción

■ En esta unidad las claves principales son:

- ▷ Entidades
- ▷ Relaciones
- ▷ Claves
- ▷ Dependencias funcionales
- ▷ Formas normales

2

FASES DEL DISEÑO

2. Fases del diseño

- La creación de una base de datos es un proceso complejo que parte de la necesidad de almacenar información del mundo real.
- Un buen diseño de una base de datos va a dar un mayor rendimiento, mejor velocidad de acceso a los datos, eliminar redundancias, y en general un mayor aprovechamiento de los recursos de los que disponemos.
- Es necesario de una metodología lo suficientemente potente para realizar esta transformación de una forma metódica y eficiente.

2. Fases del diseño

- Sin esta metodología podemos llegar a un diseño inapropiado de nuestra base de datos, que derivar en alguno de estos problemas.
 - ▷ **Redundancias:** Repetición innecesaria de información.
 - ▷ **Incoherencia:** Al existir redundancias, pueden existir incoherencias. Una depende de la otra.
 - ▷ Ej. Tenemos una base de datos con elementos duplicados, y modificamos uno de sus valores en una de las copias. Esto produce incoherencia. Pues el mismo datos esta en un lado con un valor y en otro con uno diferente.

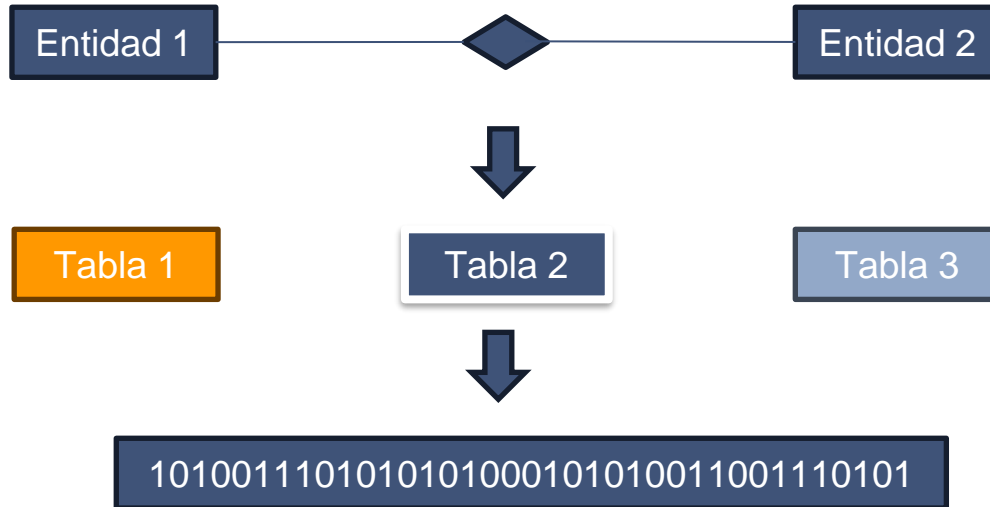
2. Fases del diseño

- Estos problemas parecen justificar por si mismos la adopción de una metodología para el diseño de la base de datos. Esta metodología se puede dividir en las siguientes fases:



2. Fases del diseño

■ Esta idea se puede resumir en el siguiente esquema:



3

MODELO ENTIDAD RELACIÓN

3. Modelo Entidad Relación

- Este modelo fue desarrollado por Peter P. Chen en 1976 en su obra “The Entity/Relationship Model”. Es el modelo conceptual mas extendido en todo el mundo.
- Su utilidad se basa en llegar a un modelo de abstracción , que permite definir los elementos que componen nuestra base de datos, de forma que sean independientes de la implementación de la misma.
- Este modelo, se compone de 2 elementos:
 - ▷ **Entidades (E) y Relaciones (R)**
 - ▷ Ambos elementos contienen **atributos**.

3.1. Entidades

- Las **entidades** se representan con un rectángulo, dentro del mismo situamos el nombre de la entidad, que el objeto sobre el que almacenamos información.
- Dentro de las entidades distinguimos 2 tipos:
 - ▶ **Entidades Fuertes.** Su existencia no depende de ninguna otra entidad.

Socios

3.1. Entidades

- ▶ **Entidades Débiles:** su existencia esta condicionada a la aparición de otra entidad. Se representa mediante un doble recuadro.
- ▶ Ej. Cuenta Bancaria y Socio. Si no hay socio no hay cuenta bancaria.

C. Bancaria

3.2. Relaciones

- Las **relaciones** se representan mediante un rombo; dentro de el situamos el nombre de la entidad.
- Las relaciones representan asociaciones entre entidades.
- Ej. Estamos navegando por Amazon Prime, y queremos alquilar una película que no esta en el catálogo. Entonces nos encontramos con 2 entidades, socios y película, y entre ellos existe una relación que es alquilan.



3.2. Relaciones

- El **grado** de una relación nos indica el conjunto de entidades que asocia.
- Una interrelación puede ser binaria (asocia 2 entidades), terciaria (asocia 3 entidades), ...
- Otro factor que tenemos que tener en cuenta son las ocurrencias de cada entidad dentro de la relación.
 - **1:1. De una a una.** Por cada ocurrencia de una entidad puede aparecer uno de la entidad asociada.
 - Ej. Vehículo y Matricula. Un vehículo solo puede tener una matricula y una matricula solo va asociada a un vehículo.

3.2. Relaciones

- ▶ **1:N. De una a muchas.** Por una ocurrencia de una entidad pueden aparecer muchas ocurrencias de la entidad asociada.
 - ▶ Ej. País y Ciudad. Un país puede tener muchas ciudades, y una ciudad solo pertenece a un país.
- ▶ **N:M. De muchas a muchas.** La cantidad de asociaciones de una entidad a otra es múltiple.
 - ▶ Ej. Asignatura y Estudiante. Un estudiante puede tener muchas asignaturas y una asignatura puede tener muchos estudiantes.

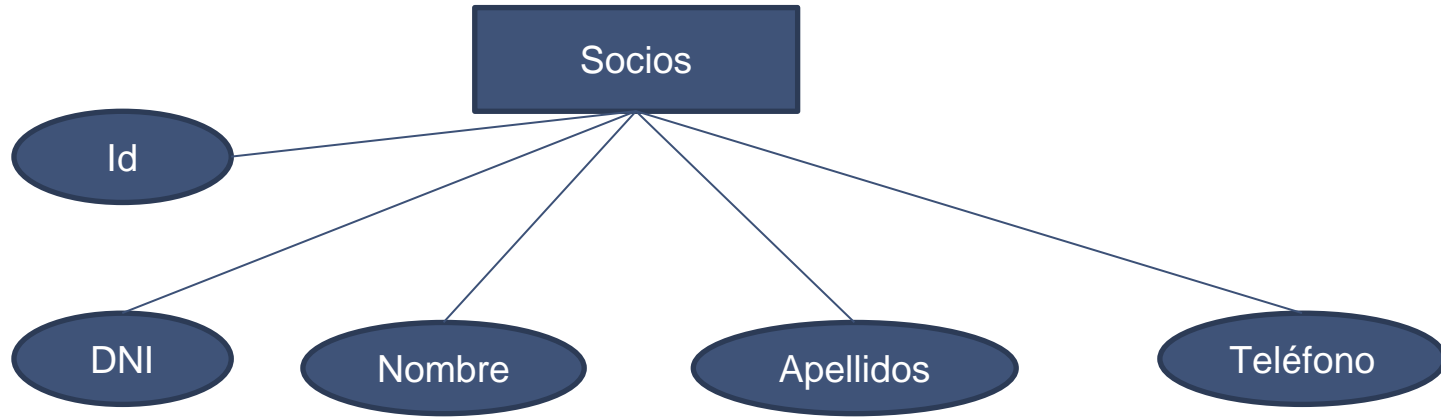
3.3. Atributos

- El ultimo elemento que puede representar el modelo entidad relación son los **atributos**. Estos almacenan las propiedades que nos interesan de las entidades.
- Los atributos se representan dentro de una elipse, y conectados mediante una línea a la entidad a la que pertenecen.
- Podemos destacar 2 tipos de clases:
 - **Atributos normales:** Dan información.
 - **Claves:** Identificar las tuplas de datos.

3.3. Atributos

- El ultimo elemento que puede representar el modelo entidad relación son los **atributos**. Estos almacenan las propiedades que nos interesan de las entidades. Los atributos se representan dentro de una elipse, y conectados mediante una línea a la entidad a la que pertenecen.
- Podemos destacar 2 tipos de clases:
 - **Atributos normales:** Dan información.
 - **Claves:** Identificar las tuplas de datos.
 - Clave Primaria: Es la clave que identifica la Entidad
 - Clave Foránea: Establece relaciones entre entidades.

3.3. Atributos

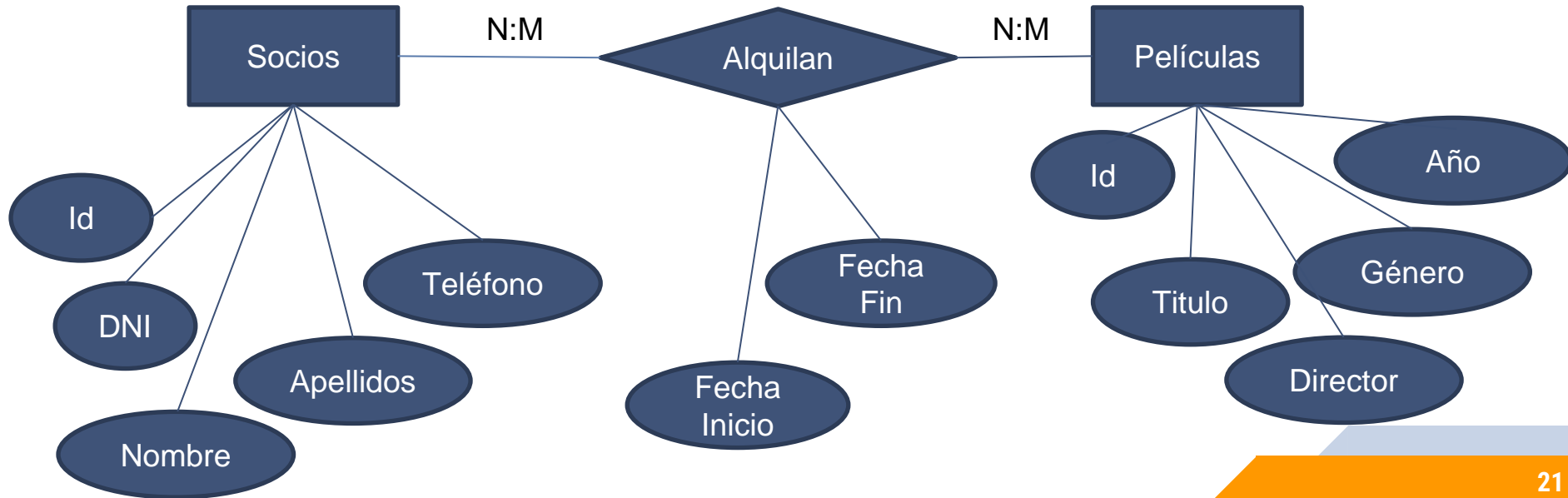


3.4. Entidades y relaciones

- Al principio es un error bastante común confundir las entidades con las relaciones.
- La diferencia entre entidades y relaciones, es que la entidades tienen existencia por si mismas, y las relaciones depende de su existencia de al menos 2 entidades.

3.4. Entidades y relaciones

■ Ejemplo Esquema Entidad – Relación Completo.



3.5. Del E-R al Modelo Lógico

- Una vez que tenemos nuestro modelo E-R completado, el siguiente paso es transformarlo a nivel lógico.
- Algunas reglas que debemos seguir:
 - Las entidades y relaciones se transforman en tablas.
 - Las relaciones tendrán una clave primaria, y claves foráneas correspondientes a las entidades con las que se relaciona.
 - Los atributos se transforman en campos de las tablas.

3.5. Del E-R al Modelo Lógico

■ El ejemplo anterior en Modelo Lógico sería:

Entidad Socios se transforma en Tabla Socios	
Id	Clave Primaria – INT
Nombre	TEXT
Apellidos	TEXT
DNI	TEXT
Teléfono	INT

3.5. Del E-R al Modelo Lógico

Entidad Películas se transforma en Tabla Películas	
Id	Clave Primaria – INT
Titulo	TEXT
Director	TEXT
Genero	TEXT
Año	INT

3.5. Del E-R al Modelo Lógico

Relación Alquilan se transforma en Tabla Alquilar	
Id	Clave Primaria – INT
Id_Socio	Clave Foránea - INT
Id_Pelicula	Clave Foránea - INT
Fecha_Inicio	TEXT o DATE
Fecha_Fin	TEXT o DATE

Ejercicios Clase

- **Ejercicio 1.** Se desea diseñar una base de datos para almacenar y gestionar la información empleada por una empresa dedicada a la venta de automóviles, teniendo en cuenta los siguientes aspectos. La empresa dispone de una serie de coches para su venta.
- Se necesita conocer la matrícula, marca y modelo, el color y el precio de venta de cada coche.
- Los datos que interesa conocer de cada cliente son el NIF, nombre, dirección, ciudad y número de teléfono: además, los clientes se diferencian por un código interno de la empresa que se incrementa automáticamente cuando un cliente se da de alta en ella. Un cliente puede comprar tantos coches como desee a la empresa. Un coche determinado solo puede ser comprado por un único cliente.

Ejercicios Clase

■ El concesionario también se encarga de llevar a cabo las revisiones que se realizan a cada coche. Cada revisión tiene asociado un código que se incrementa automáticamente por cada revisión que se haga. De cada revisión se desea saber si se ha hecho cambio de filtro, si se ha hecho cambio de aceite, si se ha hecho cambio de frenos u otros. Los coches pueden pasar varias revisiones en el concesionario”.

4

FORMAS NORMALES

4. Formas Normales

- En la segunda fase del diseño de base de datos es donde debemos emplear la teoría de la **normalización**, esta teoría no es mas que una serie de reglas que se encargan de verificar la calidad de nuestro diseño.
- Esta teoría se basa en cinco reglas que debe cumplir nuestro diseño, si queremos que este normalizado.
 - ▶ Por ejemplo, se dice que un diseño esta en tercera forma normal si cumple las 3 primeras reglas.

4. Formas Normales

- Estas reglas sirven para evitar posibles problemas que podríamos tener al actualizar las tablas. Cada forma normal elimina un tipo de redundancia.
- A medida que vamos avanzando en el diseño, las tablas se van fragmentando. Esto, lógicamente, afecta al rendimiento de la base de datos. Por lo tanto hay que encontrar un equilibrio entre la normalización y el rendimiento

4.1. Primera forma normal

- Se dice que una base de datos esta en **primer forma normal** si un atributo no puede tomar mas de un valor. Esta restricción no hace falta comprobarla, porque en el modelo de datos relacionales, cada atributo contiene un valor único.
- Ejemplo: Una cuenta bancaria contiene 2 titulares.

Banco	Nombre_Banco	Sucursal	Cuenta	Titular
0316	La Caixa	8742	123456678843	Loreto Pelegrin Alex Rodríguez

4.1. Primera forma normal

■ La forma de solucionar esto, es duplicar las tuplas:

Banco	Nombre_Banco	Sucursal	Cuenta	Titular
0316	La Caixa	8742	123456678843	Loreto Pelegrin
0316	La Caixa	8742	123456678843	Alex Rodríguez

4.2. Segunda forma normal

- Una relación esta en **segunda forma normal**:
 - ▶ Si esta en primera forma normal.
 - ▶ Si las entidades tienen su **clave primaria correspondiente**. Esa clave tiene que ser única para cada entidad.
 - ▶ Ejemplo: En un banco hay muchos socios, para crear la base de datos, se ha de elegir la clave primaria correcta. Si se escoge el nombre, puede haber mas de una persona con el mismo nombre, así que la clave primaria correcta seria un id o el DNI.

4.3. Tercera forma normal

- La **tercera forma normal** consiste en establecer las relaciones de dependencia de manera correcta.
 - ▶ Ejemplo: Supongamos que usamos una relación para almacenar datos en una cuenta.
 - ▶ Cuenta(DNI_Titular, Banco, Sucursal, Num_Cuenta, Nombre_Banco)
 - ▶ Las dependencias que encontramos son;
 - ▶ DNI-> DNI_Titular, Banco, Sucursal, Num_Cuenta
 - ▶ Banco -> Banco, Nombre_Banco

4.3. Tercera forma normal

- La forma de solucionar esto, crear otra tabla para que no haya duplicidad de datos con el banco y el nombre del Banco:

Banco	Nombre_Banco	Sucursal	Cuenta	DNI_Titular
0316	La Caixa	8742	123456678843	48660787D
0316	La Caixa	3276	884386535679	77845279K
0316	La Caixa	3436	653452213468	64234521L

4.3. Tercera forma normal

■ Solución:

Banco	Sucursal	Cuenta	DNI_Titular
0316	8742	123456678843	48660787D
0316	3276	884386535679	77845279K
0316	3436	653452213468	64234521L

Banco	Nombre_banco
0316	La Caixa

4.4. Cuarta y Quinta formas normales

- La 4 Forma Normal y la 5 Forma Norma son niveles de normalización avanzados que se utilizan para garantizar una alta integridad y consistencia de los datos. Sin embargo, su aplicación debe evaluarse cuidadosamente en función de los requisitos específicos de cada proyecto.

Ejercicios Clase

- Realizar la normalización hasta la 3º Forma Normal y explica brevemente que se ha hecho.
- **Ejercicio 1:** Una tienda en línea vende productos de diferentes categorías. Cada producto tiene un precio, un inventario y está asociado a una categoría. Los clientes pueden realizar múltiples pedidos y cada pedido puede contener varios productos.

Tabla inicial (no normalizada):

Id	Cliente	Producto	Precio	Cantidad	Categoría
1	Juan Pérez	Camiseta	19.99	2	Ropa
1	Juan Pérez	Pantalón	39.99	1	Ropa
2	María López	Libro	14.99	1	Libros



¡Gracias!