

UD 03

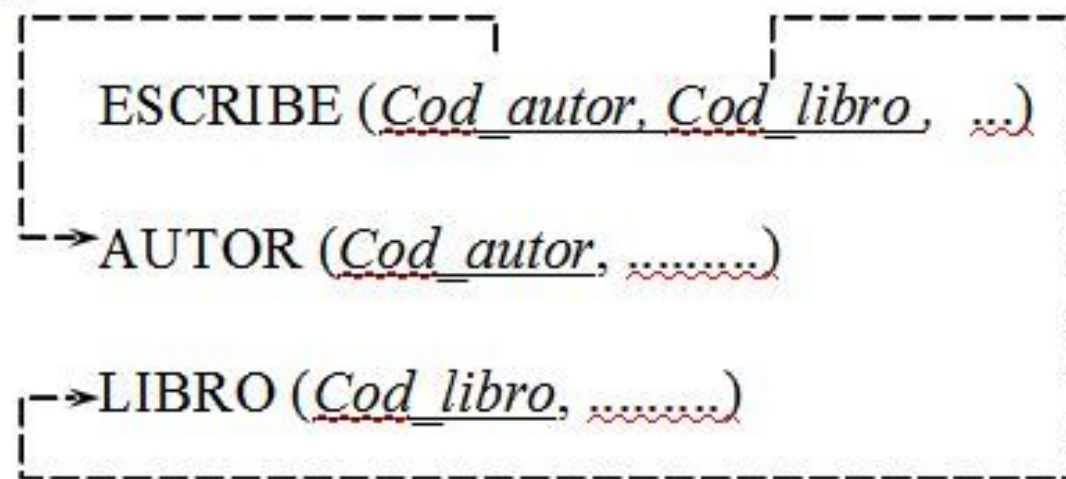


Modelo Relacional. Normalización

ME/R



MR



Contenido del tema

- Diseño lógico
- Transformación de entidades
- Transformación de atributos
- Transformación de interrelaciones
- Transformación de tipos y subtipos
- Normalización
- Ingeniería Inversa desde Workbench

Contenido del tema

- **Diseño lógico**
- Transformación de entidades
- Transformación de atributos
- Transformación de interrelaciones
- Transformación de tipos y subtipos
- Normalización
- Ingeniería Inversa desde Workbench

Diseño lógico

En el diseño lógico se pretende construir y validar el modelo lógico de los datos. Esta fase posterior a la construcción de un modelo conceptual tiene el objetivo de determinar las relaciones, validarlas tras procesos de normalización, analizar las restricciones de integridad repasando el modelo con el cliente y combinar los modelos desde una perspectiva global. Por ellos trataremos los siguientes elementos:

- Elementos y notación del modelo relacional
- Restricciones en el modelo relacional
- Notación en el diseño lógico

Diseño lógico

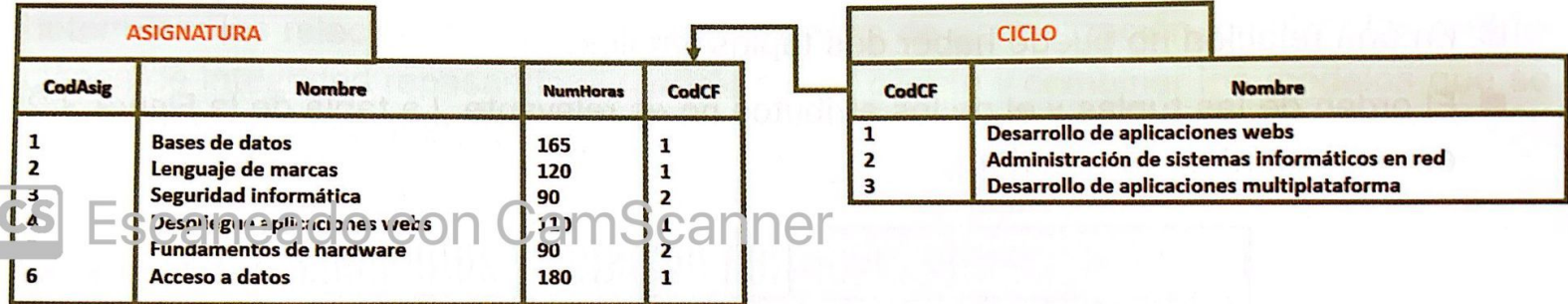
Elementos y notación del modelo relacional: Las relaciones representan en el diseño lógico lo que las entidades en el diseño conceptual. Son tablas bidimensionales en las que se almacenan información.

ASIGNATURA			
CodAsig	Nombre	NumHoras	CodCF
1	Base de datos	165	1
2	Lenguaje de marcas	120	1
3	Seguridad informática	90	2
4	Despliegue aplicaciones webs	110	1
5	Fundamentos de hardware	90	2
6	Acceso a datos	180	1

Diseño lógico

Restricciones del modelo relacional:

- En una relación no puede haber dos tuplas iguales.
- El orden de las tuplas y de los atributos no es relevante.
- Cada atributo sólo puede tomar un valor único del dominio.
- Ningún atributo que forme parte de la clave puede tomar valor nulo.



Diseño lógico

Notación en el diseño lógico: En esta etapa se obtiene un grafo compuesto por un conjunto de relaciones que interactúan entre ellas a través de sus claves ajenas (elemento fundamental de una base de datos relacional).

- Cada atributo clave primaria se subraya
- Cada atributo clave ajena se subraya **discontinuo** y además se enlaza con la relación atributo al que hace referencia.
- Cada atributo obligatorio se marca con una **N-delante**
- Cada atributo único se marca con una **U-delante**

Contenido del tema

- Diseño lógico
- **Transformación de entidades**
- Transformación de atributos
- Transformación de interrelaciones
- Transformación de tipos y subtipos
- Normalización
- Ingeniería Inversa desde Workbench

Transformación de entidades

Una vez vista la notación que se utilizará en la transformación de entidades vamos a proceder a establecer cómo se transformarán cada una de las entidades que encontramos en el modelo conceptual. Es la parte más sencilla:

A light gray rectangular box with the word "CLIENTE" centered inside in a black, uppercase, sans-serif font.

CLIENTE

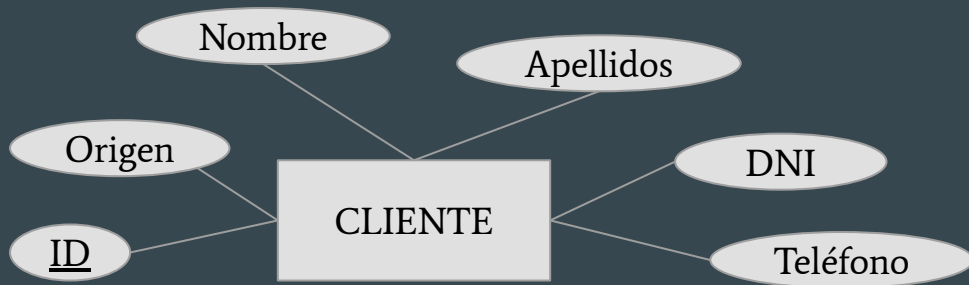
CLIENTE (clave primaria, resto de atributos....)

Contenido del tema

- Diseño lógico
- Transformación de entidades
- **Transformación de atributos**
- Transformación de interrelaciones
- Transformación de tipos y subtipos
- Normalización
- Ingeniería Inversa desde Workbench

Transformación de atributos

Los atributos de cada una de las entidades van relacionados con las propias entidades transformadas a relaciones a las que pertenecen, aplicando la notación. Esto quiere decir:



CLIENTE (ID, N-Nombre, Apellidos, U-DNI, Teléfono)

Contenido del tema

- Diseño lógico
- Transformación de entidades
- Transformación de atributos
- **Transformación de interrelaciones**
- Transformación de tipos y subtipos
- Normalización
- Ingeniería Inversa desde Workbench

Transformación de interrelaciones

Existen reglas distintas para transformar cada una de las interrelaciones que conocemos. Vamos a tratar los siguientes casos:

- **Interrelaciones N:M:** Son las más sencillas de transformar pues siempre se realiza la misma operación. Se crea una relación intermedia con las claves de la o las entidades que intervienen y los atributos relación.
- **Interrelaciones 1:N:** En la mayoría de casos se resuelve de la misma forma. Se realiza la “propagación” desde el lado del 1 al lado del N junto a los atributos relación que encuentre en su camino
- **Interrelaciones 1:1:** Son las más complejas de transformar pues se pueden resolver de distinta forma en función del origen del enunciado. Tabla intermedia o propagación. Todo depende de la obligatoriedad de las cardinalidades.

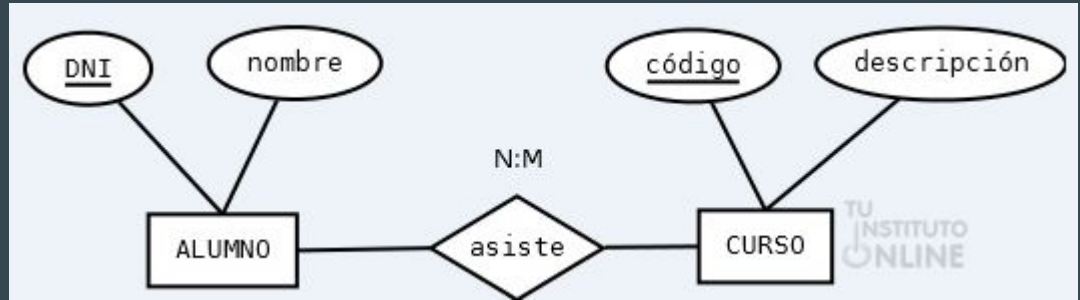
Transformación de interrelaciones

Interrelaciones N:M: Son las más sencillas de transformar pues siempre se realiza la misma operación. Se crea una relación intermedia con las claves de la o las entidades que intervienen y los atributos relación.

ALUMNO (DNI, nombre)

CURSO (código, descripción)

ASISTE (DNI, código)



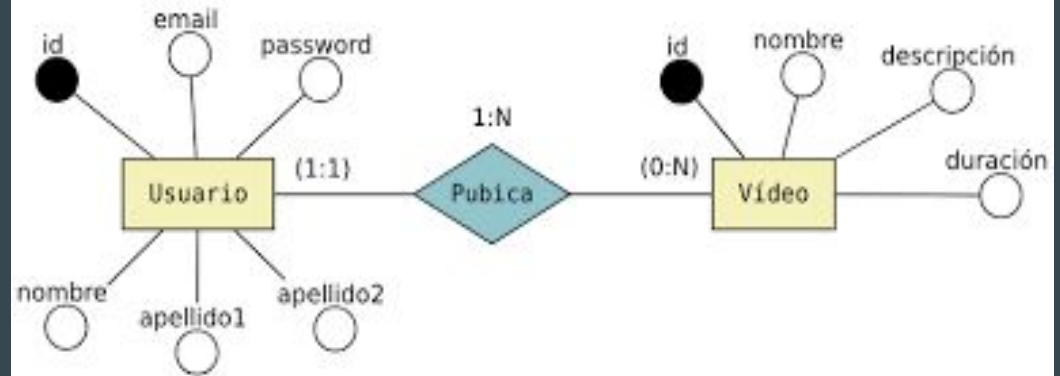
Transformación de interrelaciones

Interrelaciones 1:N: En la mayoría de casos* se resuelve de la misma forma. Se realiza la “propagación” desde el lado del 1 al lado del N junto a los atributos relación que encuentre en su camino

USUARIO(id, email, nombre ...)

VIDEO(id, nombre, ... id_usuario)

FK(id_usuario -> USUARIO)



Transformación de interrelaciones

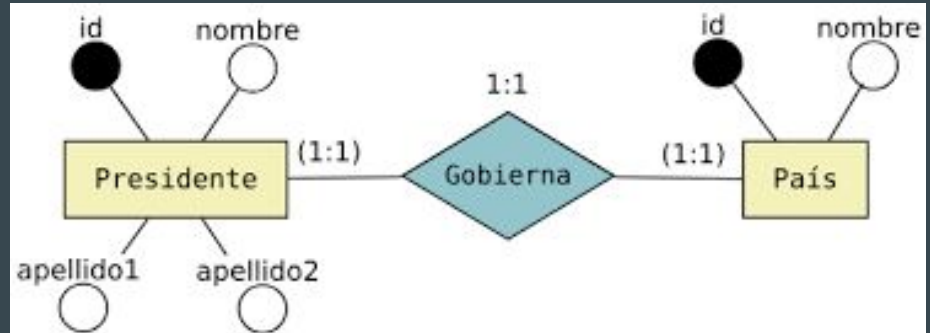
Interrelaciones 1:1: Son las más complejas de transformar pues se pueden resolver de distinta forma en función del origen del enunciado. Tabla intermedia o propagación. Todo depende de la obligatoriedad de las cardinalidades.

Ejemplo 1

PRESIDENTE(id, nombre, apellidos)

PAIS (id,nombre, id_presidente)

FK (id_presidente -> PRESIDENTE)



Transformación de interrelaciones

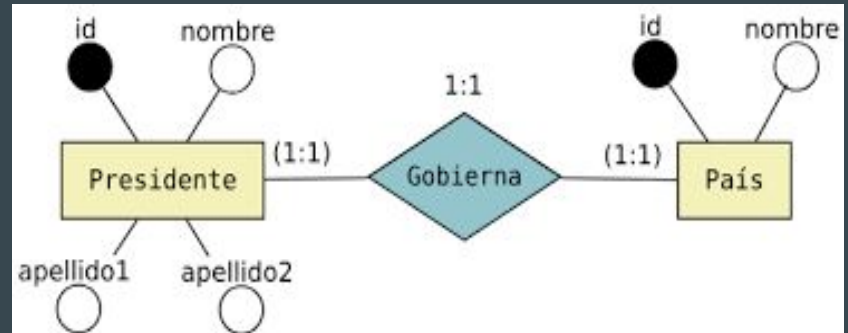
Interrelaciones 1:1: Son las más complejas de transformar pues se pueden resolver de distinta forma en función del origen del enunciado. Tabla intermedia o propagación. Todo depende de la obligatoriedad de las cardinalidades.

Ejemplo 2

PRESIDENTE(id, nombre, apellidos, id_pais)

FK(id_pais -> PAIS)

PAIS (id,nombre)



Transformación de interrelaciones

Interrelaciones 1:1: Son las más complejas de transformar pues se pueden resolver de distinta forma en función del origen del enunciado. Tabla intermedia o propagación. Todo depende de la obligatoriedad de las cardinalidades.

Ejemplo 3

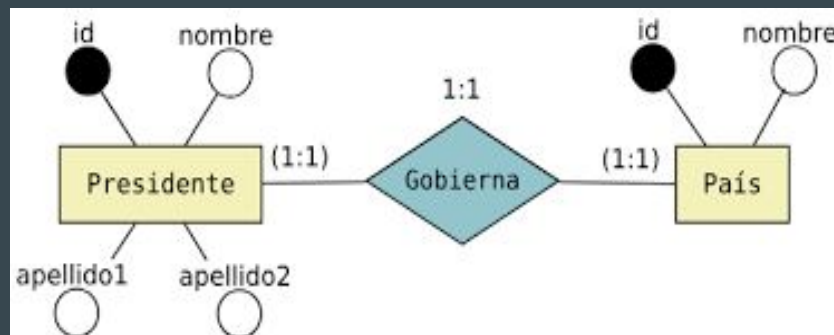
PRESIDENTE(id, nombre, apellidos, id_pais)

PAIS (id,nombre)

GOBIERNO (id_pais, id_presidente)

FK1 (id_pais -> PAIS)

FK2 (id_presidente -> PRESIDENTE)



Transformación de interrelaciones

Interrelaciones 1:1: Son las más complejas de transformar pues se pueden resolver de distinta forma en función del origen del enunciado. Tabla intermedia o propagación. Todo depende de la obligatoriedad de las cardinalidades.

¿Y si un país no estuviese obligado a tener presidente?

Transformación de interrelaciones

Hay casos concretos y específicos en el proceso de transformación de interrelaciones.

- Todos los ejemplos anteriores han sido realizados y desarrollados para relaciones binarias.
- En el caso de las reflexivas atenderemos a la misma deducción, si se trata de 1:N propagamos la clave, si se trata de N:M nueva relación y en 1:1 se entrará a valorar
- En el caso de ternarias siempre se resuelve obteniendo una nueva relación. Esto facilitará el trabajo de la implementación de nuestra base de datos.
- La peculiaridad en la relación fuerte/débil es que la clave primaria en la débil está formada por ambas claves y se suele colocar en orden pk_fuerte, pk_debil.

Contenido del tema

- Diseño lógico
- Transformación de entidades
- Transformación de atributos
- Transformación de interrelaciones
- **Transformación de tipos y subtipos**
- Normalización
- Ingeniería Inversa desde Workbench

Transformación de tipos y subtipos

Aquí entra en juego también nuestra visión global de los requisitos atendiendo a los siguientes casos:

- Si los subtipos tienen pocos atributos: Solo se crea la relación padre
- Si los subtipos tienen muchos atributos: Solo se crean las relaciones hijas
- Si ambos tienen muchos atributos se crean todos y se relacionan entre sí como una 1:N

Realiza un ejemplo de cada

Contenido del tema

- Diseño lógico
- Transformación de entidades
- Transformación de atributos
- Transformación de interrelaciones
- Transformación de tipos y subtipos
- **Normalización**
- Ingeniería Inversa desde Workbench

Normalización

El proceso de normalización en base de datos conlleva aplicar un conjunto de técnicas para convertir un esquema relacional en otro que, representando la misma información es más eficiente al contener menos redundancias y evitar anomalías en los procesos de manipulación de la información.

Se aplican técnicas basadas en teorías de conjunto, utilizando las mismas normas escritas o nomenclaturas que para el modelo relacional.

Existen 3 elementos fundamentales que tenemos que conocer para su correcta aplicación:

- Atributo multivaluado: Atributo que puede tomar más de un valor al mismo tiempo
- Dependencia funcional completa: Cada uno de los atributos que componen una relación dependen de forma completa de la clave primaria. Es decir, conociendo la clave podemos conocer el resto
- Dependencia funcional transitiva: Cuando un atributo aunque dependa de la clave, puede depender de forma transitiva o intermedia de otro.

Normalización

Atributo multivaluado: Atributo que puede tomar más de un valor al mismo tiempo.

Ejemplo: Se quiere almacenar para un cliente N teléfonos.



El campo teléfonos de la entidad cliente, para un mismo registro u ocurrencia puede tomar más de un valor. Cliente: Juan, Teléfonos: 608012238, 933212982

Normalización

Dependencia funcional completa: Cada uno de los atributos que componen una relación dependen de forma completa de la clave primaria. Es decir, conociendo la clave podemos conocer el resto y viceversa.

Sabiendo que un cliente puede tener N Facturas. Si tenemos:

Cliente(ID, nombre, apellidos, teléfono)

Cada atributo no clave depende de forma exclusiva del ID

Factura(fecha, cliente, producto, cantidad, precio, nombre cliente)

Nombre cliente no depende completamente de la clave primaria, sólo del cliente.

Normalización

Dependencia funcional transitiva: Parecido a la anterior con la peculiaridad de que existen campos que dependen funcionalmente de otros que no forman parte de la clave.

Cliente(ID, nombre, apellidos, teléfono)

Cada atributo no clave depende de forma exclusiva del ID

Empleado(ID, nombre, dni, población, provincia)

La provincia depende de la población, con lo cuál existe una dependencia funcional transitiva, entre ID->población-> provincia)

Normalización

El proceso de normalización busca solventar errores acarreados desde el diseño conceptual. Se dice que un diseño esta normalizado si cumple las siguientes reglas:

1FN: No existen atributos multivaluados.

2FN: Está en 1FN y existe dependencia funcional completa (DFC)

3FN: Esta en 2FN y no existen dependencias funcionales transitivas (DFT)

Solo si un modelo está en 3FN, nuestro modelo estará normalizado.

Contenido del tema

- Diseño lógico
- Transformación de entidades
- Transformación de atributos
- Transformación de interrelaciones
- Transformación de tipos y subtipos
- Normalización
- **Ingeniería Inversa desde Workbench**

Ingeniería inversa desde Workbench

Algunos sistemas gestores de bases de datos nos permiten obtener una versión gráfica del modelo relacional o grafo-relacional dado esquemas de bases de datos físicos ya implementados. Esto es muy útil a la hora de trabajar con nuestras bases de datos.

En concreto, My SQL Workbench nos permite obtenerlos dirigiéndonos a la opción

Database -> Reverse Engineer -> Select Schema -> Execute

