DISEÑO LÓGICO DE BASES DE DATOS - MODELO RELACIONAL

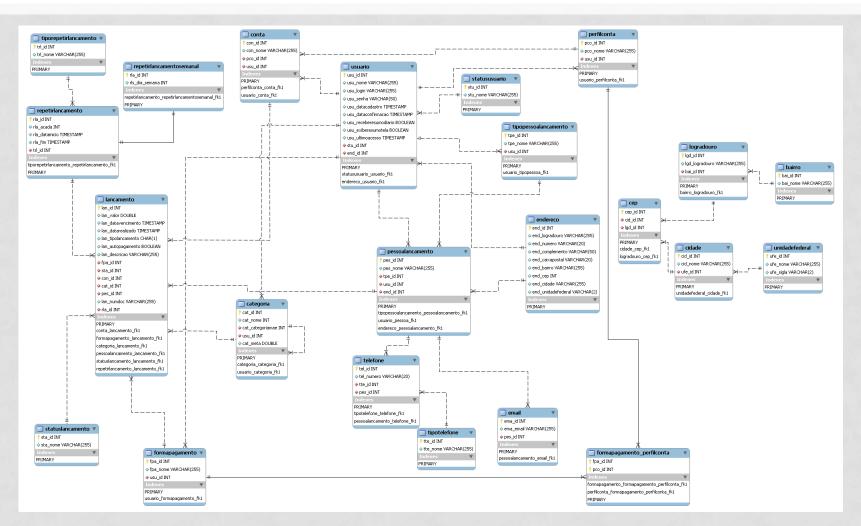
ÍNDICE

- 1. El modelo relacional.
- 2. Transformación E/R al modelo relacional.
- 3. Normalización.

1. EL MODELO RELACIONAL

- 1. Las relaciones en el modelo relacional
- 2. Otros conceptos del modelo relacional

1. EL MODELO RELACIONAL



1. EL MODELO RELACIONAL

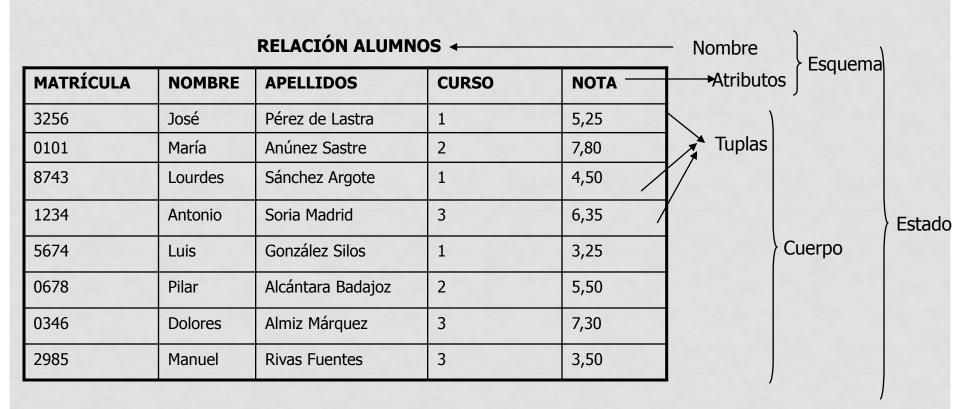
- El objetivo principal del modelo relacional es proteger al usuario de la obligación de conocer las estructuras de datos físicas con las que se representa la información de una BBDD, así se permite que la BBDD pueda implementarse en cualquier gestor de BBDD relacional (SQL). Las características de este modelo son:
 - La relación es el elemento fundamental del modelo. Estas relaciones se pueden operar mediante el Álgebra Relacional.
 - El modelo relacional es independiente de la forma en que se almacenan los datos y de la forma de representarlos, por tanto la BBDD se puede representar en cualquier SGBD

•

1.1. LAS RELACIONES EN MOD. RELACIONAL

- Se define una **relación** como un conjunto de atributos, cada uno de los cuales pertenece a un dominio, y que posee un nombre que identifica la relación. Se representa gráficamente por una tabla con columnas (**atributos**) y filas (tuplas). El conjunto de tuplas de una relación representa el **cuerpo** de la relación y el conjunto de atributos y el nombre representan el **esquema**.
- Actualmente los modelos lógicos más extendidos con diferencia son el modelo relacional y los diagramas de clases que utiliza UML para modelar las BBDD orientadas a objetos. El modelo relacional de Codd se ajusta a la perfección al modelo E/R de Chen.

1.1. LAS RELACIONES EN MOD. RELACIONAL



- Vamos a definir los conceptos necesarios para transformar el modelo conceptual (diagrama E/R) en el modelo lógico (modelo relacional).
 - Atributos: características que describen a una entidad o relación.
 - Dominio: conjunto de valores permitidos para un atributo. Por ejemplo, cadena de caracteres, número enteros, los valores Sí o No, etc.
 - Restricciones de semántica: condiciones que deben cumplir los datos para su correcto almacenamiento. Hay varios tipos:
 - <u>Restricciones de clave</u>: es el conjunto de atributos que identifican de forma única a una entidad.

- Restricciones de valor único (UNIQUE): impide que un atributo tenga un valor repetido. Todos los atributos clave cumplen esta restricción. Puede ser que algún atributo que no sea clave sea necesaria esta restricción, por ejemplo, el número de bastidor de un coche, que no es clave principal, lo es matrícula, no se puede repetir.
- <u>Restricciones de integridad referencial</u>: se da cuando una tabla tiene una referencia a algún valor de otra tabla. En este caso la restricción exige que exista el valor referenciado a la otra tabla. Por ejemplo, no se puede poner una nota a un alumno que no exista.
- <u>Restricciones de dominio</u>: exige que el valor que puede tomar un campo esté dentro del dominio definido. Por ejemplo, si se establece que un campo DNI pertenece al dominio de los números de 9 dígitos + 1 letra no es posible insertar un DNI sin letra.

- <u>Restricciones de verificación (CHECK)</u>: permiten comprobar si un valor de un atributo es válido conforme a una expresión.
- Restricción de valor NULO (NULL o NOT NULL): un atributo puede ser obligatorio si no admite el valor NULO o NULL. Si admite como valor el valor NULL el atributo es opcional.
- <u>Disparadores o triggers</u>: son procedimientos que se ejecutan para hacer una tarea concreta en el momento de insertar, modificar o eliminar información de un tabla.
- Restricciones genéricas adicionales o aserciones (ASSERT): permite validar cualquiera de los atributos de una o varias tablas.

- Clave: una clave es un atributo o conjunto de atributos que identifican de forma única una ocurrencia de entidad. Las claves pueden ser simples (atómicas) o compuestas. Hay varios tipos de claves:
 - <u>Superclave</u>: identifican a una entidad. Por ejemplo, para un empleado, las superclaves posibles son el DNI, o el DNI+Nombre o el DNI+Nombre+Número de la Seguridad Social, etc.
 - Clave Candidata: La clave candidata de una relación es el conjunto de uno o más atributos que determinan unívoca y mínimamente cada tupla. Siempre hay una clave candidata pues por definición no puede haber dos tuplas iguales En una relación pueden existir varias claves candidatas.

- EMPLEADOS (CÓDIGO, DNI, DIRECCIÓN, TFNO, SUELDO, COMISIÓN)
 - En esta relación podrían ser claves candidatas tanto **código** como **DNI.** Ambas son claves candidatas.
 - <u>Clave Primaria</u>: es la clave candidata elegida por el diseñador como clave definitiva, en el ejemplo anterior se elegiría Código y se representa subrayada. A la clave candidata no elegida como clave primaria se le llama clave alternativa.
 - Clave Ajena o Foránea: es un atributo de una entidad, que es clave en otra entidad.
 - Por ejemplo,
 - EMPLEADOS (<u>COD-EMP</u>, DNI, NOMBRE, TFNO, COD-DEP)
 - DEPARTAMENTOS (COD-DEP, NOMBRE, LOCALIDAD)
 - COD-DEP: clave ajena que referencia a COD-DEP

Ejemplo

- EDITORIAL (cod edit, nombre, dirección, ciudad)
- LIBRO (código, título, idioma, cod_edit)
- ESCRIBE (cod_autor, cod_libro)
- AUTOR (cod autor, nombre, nacionalidad)
 - Cod_edit: clave ajena que referencia a cod_edit de la tabla EDITORIAL.
 - Cod_libro: clave ajena que referencia a código de la tabla LIBRO.
 - Cod_autor: clave ajena que referencia a cod_autor de la tabla AUTOR.

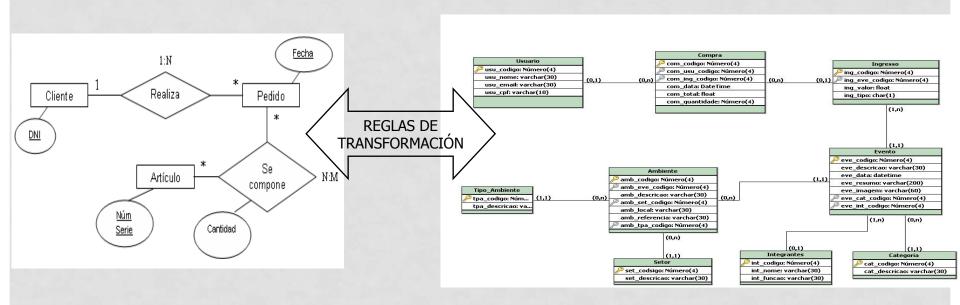
- Para las claves ajenas existe la regla de Integridad referencial que establece que los valores de la clave ajena o bien coinciden con los de la clave primaria a la que referencian o bien son nulos. Por tanto, para cada clave ajena se debe especificar si puede o no tomar valores nulos. Además, es necesario determinar las consecuencias de ciertas operaciones (borrado y modificación) realizadas sobre tuplas de la relación referenciada.
- Se puede distinguir entre:

- Operación restringida (RESTRICT): el borrado o modificación de tuplas de la relación que contienen la clave primaria referenciada sólo se permite si no existen tuplas con dicha clave en la relación que contiene la clave ajena. Ejemplo: en el caso anterior esto implicaría que sólo podemos borrar una editorial que no tenga ningún libro, de haberlo, el sistema impediría el borrado.
- Operación con transmisión en cascada (CASCADE): el borrado o modificación de tuplas de la relación que contienen la clave primaria referenciada lleva consigo el borrado o modificación en cascada de las tuplas de la relación que contienen la clave ajena. Ejemplo: en el caso anterior al modificar el código de una editorial, se debe modificar también dicho código en todos los libros de nuestra Base de Datos que sean de dicha editorial.

- Operación con puesta a nulos (SET NULL): el borrado o modificación de tuplas de la relación que contienen la clave primaria referenciada lleva consigo "poner a nulos" los valores de las claves ajenas de la relación que referencia. Ejemplo: en el caso anterior cuando se borra una editorial, a todos aquellos libros que sean de esa editorial y que se encuentran en la relación LIBROS se les colocaría el atributo cod_edit a nulo. Esta opción sólo es posible cuando el atributo que es clave ajena admite el valor nulo.
- Operación con puesta a valor por defecto (SET DEFAULT): el borrado o modificación de tuplas de la relación que contienen la clave primaria referenciada lleva consigo "poner al valor por defecto a la clave ajena de la relación que referencia; valor por defecto que habrá sido definido al crear la tabla correspondiente.

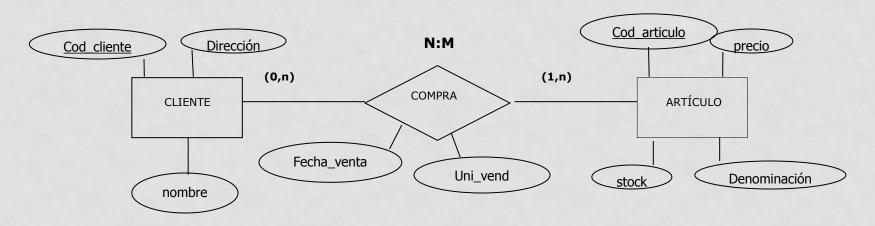
- Operación que desencadena un procedimiento de usuario: el borrado o modificación de tuplas de la relación que contienen la clave primaria referenciada lleva consigo la puesta en marcha de un procedimiento definido por el usuario.
- Las opciones seleccionadas para el borrado y modificación son independientes.

2. TRANSFORMACIÓN DE UN DIAGRAMA E/R AL MODELO RELACIONAL



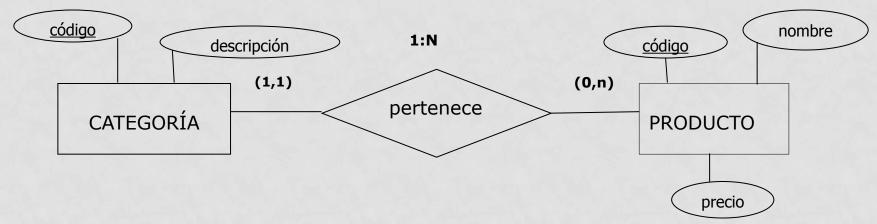
Transformación del modelo E/R en relacional

- La transformación del modelo conceptual (E/R) al modelo lógico (relacional) se basa en las siguientes reglas:
 - Toda entidad se transforma en una relación.
 - Todo atributo se transforma en columna dentro de una relación.
 - Los atributos AIP (atributo identificador principal), se convierten en Clave Primaria y los AIA (atributos identificador alternativo) en Clave alternativa.
 - Toda relación N:M se transforma en una relación que tendrá como clave primaria la concatenación de los atributos clave de las entidades que asocia.



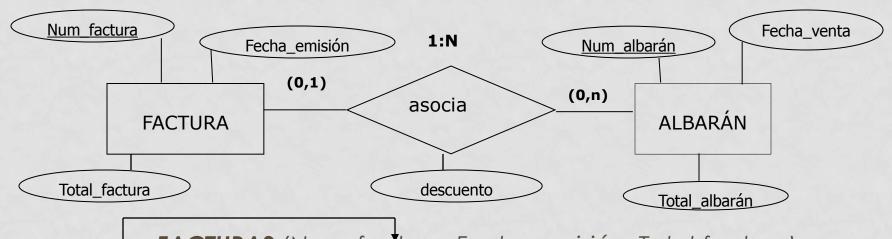
- Para el diagrama E/R, las relaciones generadas son:
 - CLIENTES (Cod_cliente, Nombre, Dirección)
 - ARTÍCULOS (Cod_articulo, Precio, Stock, Denominación)
 - COMPRAS (<u>Cod_cliente</u>, <u>Cod_articulo</u>, <u>Uni_vend</u>, Fecha_venta)

- Transformación de entidades fuertes:
 - Para cada entidad A, entidad fuerte, con atributos (a1, a2,...,an) se crea una relación A (con el nombre en plural) con n columnas correspondientes a los atributos de A, donde cada fila de la relación A corresponde a una ocurrencia de la entidad A. La clave primaria de la relación A la forman los atributos clave de la entidad A.



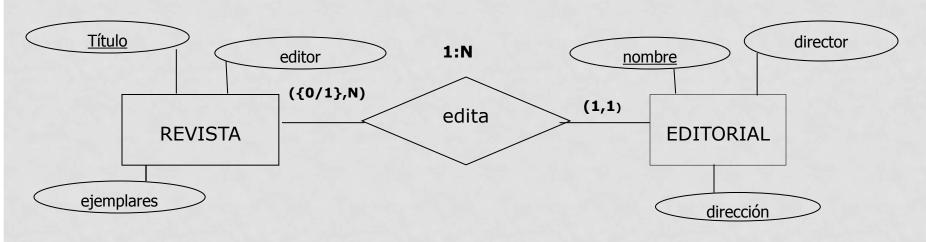
- En el diagrama E/R, las relaciones generadas son (sin tener en cuenta relaciones nuevas o claves ajenas):
 - · CATEGORÍAS (código, descripción)
 - PRODUCTOS (código, nombre, precio)

- Relaciones con cardinalidad 1:N: existen dos soluciones
 - Transformar la interrelación en una relación: se hace como si se tratara de una relación N:M. Esta solución se puede elegir cuando se prevé que en un futuro la relación se convertirá en N:M y cuando la interrelación tiene atributos propios. También se crea una nueva relación cuando la cardinalidad es opcional en el lado del 1, es decir (0,1). La clave de esta relación es la de la entidad del lado N.

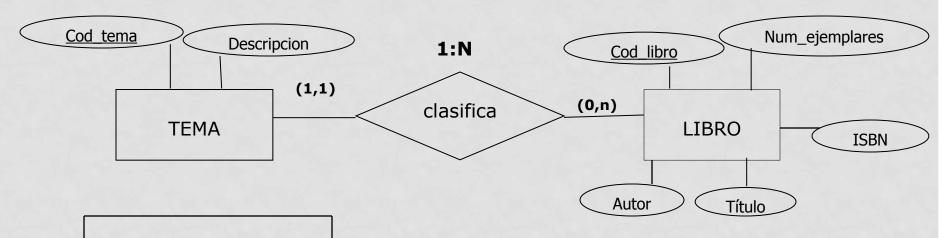


- FACTURAS (Num factura, Fecha-emisión, Total-factura)
- ALBARANES (<u>Num albarán</u>, Fecha-venta, Total-albarán)
- FAT-ALB (Num_factura, Num albarán, Descuento)
- Donde Num-albarán es la Clave Principal, y tanto Num_albaran como Num-factura son claves ajenas que referencian a sus respectivas relaciones.

- Relaciones con cardinalidad 1:N, otra solución:
 - Propagar la clave: este caso se aplica cuando la cardinalidad es obligatoria en el lado del 1, es decir, cuando tenemos cardinalidad (1,1). Se propaga el atributo principal de la entidad que tiene cardinalidad máxima 1 a la relación que genera la entidad que tiene cardinalidad máxima N. Si existen atributos propios en la relación, estos también se propagarán.

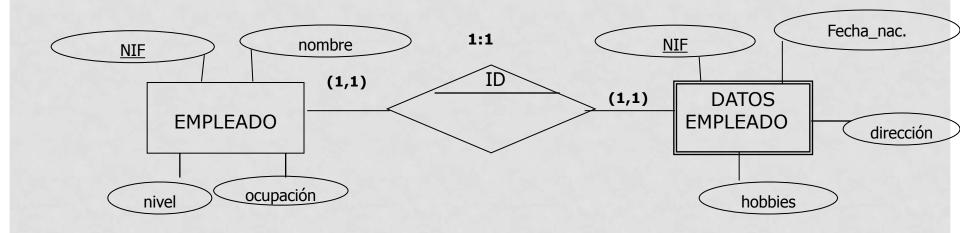


- EDITORIALES (Nombre, Director, Dirección)
- REVISTAS (<u>Título</u>, Editor, Ejemplares, Nombre_edit)
- Donde Nombre_edit es una clave ajena que referencia a la tabla EDITORIAL

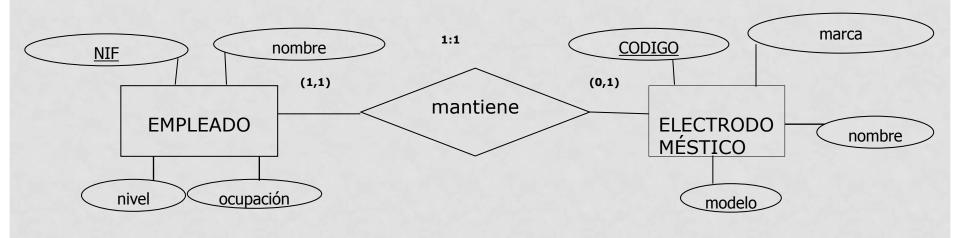


- TEMAS (Cod tema, Descripción)
- LIBROS (<u>Cod_libro</u>, Autor, ISBN, Título, Num_ejemplares,
 Cod_tema)
- Se propaga la clave, de la entidad TEMA a la entidad LIBRO.

- En la transformación de relaciones 1:1 se tiene en cuenta las cardinalidades de las entidades que participan. Existen dos soluciones:
 - Transformar la interrelación en una relación: si ambas entidades poseen cardinalidades (0,1).
 - Propagar la clave: si una de las entidades posee cardinalidad (0,1) y la otra (1,1), conviene propagar la clave de la entidad con cardinalidad (1,1) a la relación resultante de la entidad de cardinalidad (0,1). Si ambas entidades poseen cardinalidades (1,1), se puede propagar la clave de cualquiera de ellas a la tabla resultante de la otra. En este caso (1,1), también se pueden añadir los atributos de una entidad a otra, resultando una única tabla con todos los atributos de las entidades y de la relación, si los hubiera, eligiendo como clave primaria una de las dos.



- EMPLEADOS (Nif., nombre, Ocupación, Nivel, Fecha_Nac, Hobbies, Dirección)
- Donde NIF será la Clave Principal.



- EMPLEADOS (NIF, nombre, ocupación, nivel)
- ELECTRODOMÉSTICOS (Código, nombre, marca, modelo, NIF)
- Donde NIF será la C. Ajena que referencia a EMPLEADO

Ejercicios 1

 Un empleado ocupa un solo puesto de trabajo, y ese puesto de trabajo es ocupado por un solo empleado o por ninguno. De los empleados queremos guardar su código, nombre, dirección y teléfono. Del puesto de trabajo el código, descripción, oficina, despacho y mesa. Realiza el Modelo E/R y transfórmalo al Modelo Relacional.

Ejercicios 2

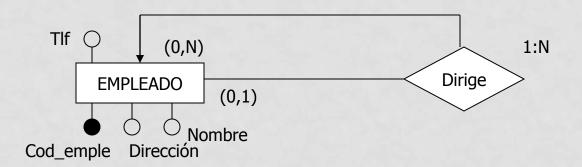
• Se desea mecanizar la biblioteca de un centro educativo. En la biblioteca existen fichas de autores y libros. Un autor puede escribir varios libros, y un libro puede ser escrito por varios autores. Un libro está formado por ejemplares que son los que se prestan a los usuarios. Así un libro tiene muchos ejemplares y un ejemplar pertenece solo a un libro. De los ejemplares nos interesa saber la localización dentro de la biblioteca, los ejemplares son prestados a los usuarios, un usuario puede tomar prestados varios ejemplares y un ejemplar puede ser prestado a varios usuarios. Del préstamo nos interesa saber la fecha de préstamo y la de devolución

- Relaciones Reflexivas o recursivas: son relaciones binarias en las que participa un tipo de entidad. En el proceso de convertir una relación reflexiva a tabla hay que tener en cuenta sobre todo la cardinalidad. Lo normal es que toda relación reflexiva se convierta en dos tablas, una para la entidad y otra para la relación. Se puede presentar los siguientes pasos:
 - Si la relación es 1:1, la clave de la entidad se repite, con lo que la tabla resultante tendrá dos veces ese atributo, una como clave primaria y otra como clave ajena de ella misma. No se crea la segunda tabla.

- Si la relación es 1:N, podemos tener dos casos:
 - Caso de que la entidad muchos sea siempre obligatoria se procede como en el caso 1:1
 - Si no es obligatoria, se crea una nueva tabla cuya clave será la de la entidad del lado muchos, y además se propaga la clave a la nueva tabla como clave ajena.
- Si es N:M se trata igual que en las relaciones binarias. La tabla resultante de la relación contendrá dos veces la clave primaria de la entidad del lado muchos, más los atributos de las relaciones si los hubiera. La clave de esta nueva tabla será la combinación de las dos.

• Ejemplo1:

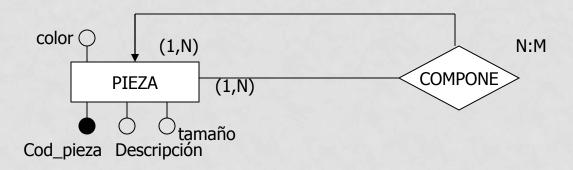
 Consideramos la relación Empleado-dirige-empleado. Un empleado puede dirigir a muchos empleados o a ninguno. Y un empleado es dirigido por un director o por ninguno si él es el que dirige. En este caso no hay obligatoriedad en la entidad muchos. El Modelo E/R es:



- La relación DIRIGE tiene como clave primaria el código de empleado, que a su vez será clave ajena. Además, se le añade el código de empleado pero, en este caso, tendrá el papel de director, que a su vez será clave ajena a la tabla EMPLEADO. El resultado será:
 - EMPLEADO (cod emple, dirección, teléfono, nombre)
 - DIRIGE (cod emple(FK), cod_direc(FK))

Ejemplo2:

 Consideramos la relación una pieza se compone de muchas piezas, que a su vez están compuestas de otras piezas, es decir Pieza_Compone_Pieza. El Modelo E/R es:

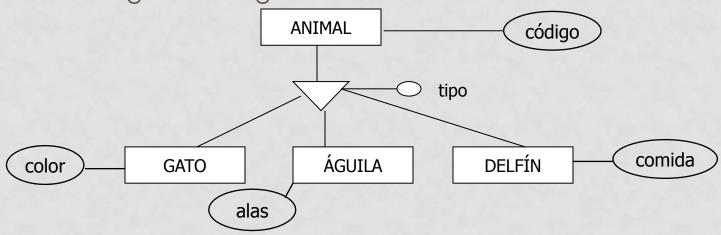


- PIEZA (cod pieza, descripción, tamaño, color)
- COMPONE_PIEZA (cod pieza comp(FK), cod pieza(FK))

Cod_pieza_comp Y cod_pieza son clave ajenas y representan cada código de pieza que esté compuesto por otras piezas (Cod_pieza_comp) y cada pieza que lo compone (cod_pieza). Ambas concatenadas constituyen la clave de la relación.

Transformación de otros elementos del modelo E-R

- Jerarquías al modelo relacional
 - El modelo relacional no dispone de mecanismos fáciles de usar que permitan la representación de relaciones jerárquicas, por lo tanto se tienen que eliminar. Para pasar estas relaciones al modelo relacional se aplicará una de las siguientes reglas:



- Integrar todas las entidades en una única eliminando los subtipos. Esta nueva entidad contendrá todos los atributos del supertipo, todos los de los subtipos, y los atributos discriminativos para distinguir a qué subentidad pertenece cada atributo.
 Todas las relaciones se mantiene con la nueva entidad. Para exclusivas:
 - ANIMALES (código, tipo, color, alas, comida)
- Eliminación del supertipo, transfiriendo los atributos del supertipo a cada uno de los subtipos. Las relaciones del supertipo se consideran para cada uno de los subtipos. Solo puede ser aplicada para jerarquías totales y exclusivas.
 - GATOS (<u>código</u>, color)
 - AGUILAS (<u>código</u>, alas)
 - DELFINES (<u>código</u>, comida)

- Insertar una relación 1:1 entre el supertipo y cada uno de los subtipos. Los atributos se mantienen y cada subtipo se identificará con la clave ajena del supertipo. El supertipo mantendrá una relación 1:1 con cada subtipo. Los subtipos mantendrán, si la relación es exclusiva, la cardinalidad mínima 0, y si es inclusiva 0 ó 1.
 - ANIMALES (código)
 - GATOS (<u>código (FK)</u>, color)
 - AGUILAS (<u>código(FK)</u>, alas)
 - DELFINES (<u>código(FK)</u>, comida)

• Transformación de otros elementos del modelo E/R

- Relaciones N-arias
 - En este tipo de relaciones se agrupan 3 o más entidades y para pasar al modelo de datos relacional cada entidad se convierte en una relación, así como la interrelación, que va a contener los atributos propios de ella más las claves de todas las entidades. La clave de la tabla resultante será la concatenación de las claves de las entidades. Hay que tener en cuenta que:
 - Si la **relación es N:N:N**, es decir, si todas las entidades participan con cardinalidad N, la clave de la tabla resultante es la unión de las claves de las entidades que relaciona. Esa tabla incluirá los atributos de la relación si lo hubiera.

Ejemplo

Cod_profesor

N:M:P

Cod_curso

Descripción

(1,N)

Imparte

Curso

Tif

Especialidad

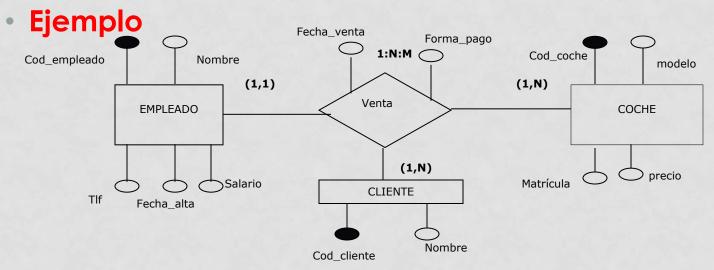
Curso

ASIGNATURA

Nivel

Turno

- PROFESORES (cod profesor, dirección, nombre, tlf, especialidad)
- CURSOS (<u>cod curso</u>, descripción, nivel, turno)
- ASIGNATURAS (Cod asignatura, nombre)
- IMPARTE (cod_profesor(FK), cod_curso(FK), cod_asignatura(FK))



- CLIENTES (<u>cod cliente</u>, nombre),
- EMPLEADO (cod empleado, tlf, salario, fecha_alta)
- COCHES (Cod coche, modelo, matrícula, precio)
- VENTA (<u>cod_coche(FK)</u>, <u>cod_cliente(FK)</u>, <u>cod_empleado(FK)</u>, forma_pago, fecha_venta)

Resumen

MODELO	MODELO ENTIDAD RELACIÓN		MODELO RELACIONAL
	ENTIDAD		TABLA
		1:N	Propagar la clave: clave ajena (tabla con N), o bien otra tabla
	DINADIAC	1:1	Propagar la clave: clave ajena, o bien otra tabla o bien uniendo las dos entidades en una tabla
Se	BINARIAS	N:M	Otra tabla nueva, con clave primaria igual a la suma de las claves primarias de las dos tablas
oue Oue	REFLEXIVA S	1:N	Clave ajena o bien otra tabla
aci		1:1	Clave ajena o bien otra tabla
Relaciones	J	N:M	Otra tabla nueva, relación con clave principal igual a la clave principal de la entidad duplicada
	TERNAF	RIAS	Se crea una tabla nueva y a la hora de elegir la clave se tendrá en cuenta: 1 Concatenación de claves primarias de las entidades, con grado diferente a 1 (N,M,P) 2 Si alguna tiene cardinalidad máxima 1, al menos ha de haber (N-1) claves primarias de otras (N-1) entidades, y han de participar en la relación las claves primarias de las entidades con cardinalidad máxima 1.

3. NORMALIZACIÓN

- 1. Teoría de la Normalización.
- 2. Noción intuitiva de las Formas Normales.
- 3. Primera Forma Normal.
- 4. Segunda Forma Normal.

- 5. Tercera Forma Normal.
- 6. Otras Formas Normales

3.1. TEORÍA DE LA NORMALIZACIÓN

- La Teoría de la normalización define una serie de reglas que debe cumplir un modelo relacional para garantizar:
 - Que no existen redundancias en la base de datos, con lo que se disminuye el espacio requerido para el almacenamiento de la información y se reducen los problemas de integridad de la información almacenada en la base de datos.
 - Que se representa de forma coherente los objetos y relaciones existentes en el sistema.
 - Que el rendimiento de las operaciones de actualización es el máximo posible.
 - Que las operaciones de consulta son fiables y su rendimiento es el máximo posible

3.1. TEORÍA DE LA NORMALIZACIÓN

- A este conjunto de reglas se les denomina Reglas de normalización de relaciones. Se dice que una relación está en una determinada forma normal si satisface un cierto conjunto específico de restricciones impuestas por la regla de normalización correspondiente.
- La sucesiva aplicación de las reglas de normalización, empezando por la Primera Forma Normal, da lugar a un mayor número de relaciones en el esquema que satisfacen dichas reglas.

Dependencias funcionales

- Son de primordial importancia a la hora de encontrar y eliminar la redundancia de los datos almacenados en las tablas de una base de datos relacional, se centran en el estudio de las dependencias que presenta cada atributo de una relación con respecto al resto de atributos de la misma.
- Dada una relación R que contiene los atributos X e Y se dice que Y depende funcionalmente de X (X → Y) si y sólo si en todo momento cada valor de X tiene asociado un solo valor de Y. Esto es lo mismo que decir que si dos tuplas de R tienen el mismo valor para su atributo X forzosamente han de tener el mismo valor para el atributo Y.

- Tipos de dependencias:
 - Dependencia Transitiva: dados los atributos X, Y y Z de la relación R en la que existen las siguientes dependencias funcionales X—→Y; Y—→Z; se dice que Z tiene una dependencia transitiva respecto a X a través de Y:X—→Z. Por ejemplo:

Nombre de alumno --> Dirección

Dirección → Ciudad

Nombre de alumno → Ciudad.

 Existen seis niveles de normalización de una relación, que se muestran en la figura. Una relación se encuentra en un grado u otro de normalización si cumple una serie de propiedades (restricciones) que se explican a continuación.

1FN	
2FN	I
3F	·N
	NBC
	4FN
	5FN

- Las tres primeras formas normales las definió Codd y se basan en las dependencias funcionales que existen entre los atributos de la relación. Más tarde, y debido a que todavía existían anomalías, redefinió la 3FN y la llamó FNBC. Posteriormente se definieron dos niveles más de normalización, la 4FN y la 5FN.
- De esta forma, las relaciones en 1FN tienen más redundancia de datos que los niveles superiores, y por lo tanto más anomalías de actualización de datos. Y la 5FN es el grado de normalización máximo que puede alcanzar una relación.

- Se dice que una relación está en 1FN si y sólo si los valores que componen cada atributo de una tupla son atómicos, es decir, cada atributo de la relación toma un único valor del dominio correspondiente, o lo que es lo mismo, no existen grupos repetitivos.
- La tabla estará en 1FN si tiene un solo valor en la intersección de cada fila con cada columna. Un conjunto de relaciones se encuentra en 1FN si ninguna de ellas tiene grupos repetitivos.
- Si una relación no está en 1FN, hay que eliminar de ellas los grupos repetitivos. Un grupo repetitivo será el atributo o grupo de atributos que tiene múltiplos valores para cada tupla de la relación. Hay dos formas de eliminar los grupos repetitivos:

- Repetir los atributos con un solo valor para cada valor del grupo repetitivo. De este modo se introducen redundancias ya que se duplican valores, pero estas redundancias se eliminarán después mediante las restantes formas normales.
- La segunda forma de eliminar los grupos repetitivos consiste en poner cada uno de ellos en una relación a parte, heredando la clave primaria de la relación en al que se encontraban.

• **Ejemplo**. Consideramos la tabla Alumno, con clave primaria Cod_Alumno, en la que el atributo Tlf puede tomar varios valores: el móvil, el de casa, el del padre, el de la madre, etc.

COD ALUMNO	NOMBRE	APELLIDO	TLF	DIRECCIÓN
1111	PEPE	GARCÍA	678-900600 91-2233441 91-1231232	C/ Las Cañas, 45
2222	MARÍA	SUÁREZ	91-7008001	C/Mayor, 12
3333	JUAN	GIL	91-7562324 660-111222	C/La plaza
4444	FRANCISCO	MONTOYA	678-556443	C/La arboleda

 Esta tabla no está en 1FN, ya que hay dos alumnos con varios tlfs

 Definir como clave primaria de la tabla Cod_Alumno y el Tlf, con el fin de que cada atributo tome un único valor en la tupla correspondiente

COD ALUMNO	<u>TLF</u>	NOMBRE	APELLIDO	DIRECCIÓN
1111	678-900600	PEPE	GARCÍA	C/ Las Cañas, 45
1111	91-2233441	PEPE	GARCÍA	C/ Las Cañas, 45
1111	91-1231232	PEPE	GARCÍA	C/ Las Cañas, 45
2222	91-7008001	MARÍA	SUÁREZ	C/Mayor, 12
3333	91-7562324	JUAN	GIL	C/La plaza
3333	660-111222	JUAN	GIL	C/La plaza
4444	678-556443	FRANCISCO	MONTOYA	C/La arboleda

Tabla alumno primera transformación a 1FN

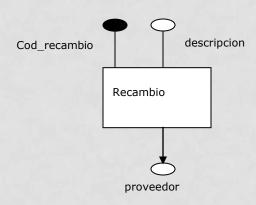
O también se eliminan los grupos repetitivos (TLF)
 y se crea una relación (tabla) junto con la clave inicial.

COD ALUMNO	NOMBRE	APELLIDO	DIRECCIÓN
1111	PEPE	GARCÍA	C/ Las Cañas, 45
2222	MARÍA	SUÁREZ	C/Mayor, 12
3333	JUAN	GIL	C/La plaza
4444	FRANCISCO	MONTOYA	C/La arboleda

COD ALUMNO (FK)	<u>TLF</u>
1111	678-900600
1111	91-2233441
1111	91-1231232
2222	91-7008001
3333	91-7562324
3333	660-111222
4444	678-556443

Tabla alumno segunda transformación a 1FN

- La aplicación de esta regla es muy sencilla, simplemente consiste en descomponer los atributos multivaluados. Estos atributos dan lugar a una nueva relación cuya clave primaria es la concatenación de la clave primaria de la entidad en la que se sitúa el atributo multivaluado más el nombre del atributo multivaluado.
- Supongamos un modelo E/R con la siguiente entidad:



- Aplicando la regla 1 obtenemos la relación:
 - RECAMBIO(cod recambio, descripción, proveedor)
 - donde proveedor es un atributo multivaluado (para un mismo recambio puedo tener varios proveedores).
- Para cumplir la 1FN realizaríamos la siguiente descomposición:
 - RECAMBIO(<u>cod recambio</u>, descripción)
 - PROVEEDOR(cod_recambio, proveedor)

- Se dice que una relación se encuentra en 2FN si y sólo si satisface la 1FN, y cada atributo de la relación que no está en la clave depende funcionalmente de forma completa de la clave primaria de la relación.
 La 2FN se aplica a las relaciones que tienen claves primarias compuestas por dos o más atributos.
- Si una relación está en 1FN y su clave primaria es simple (tiene un solo atributo), entonces también está en 2FN. Las relaciones que no están en 2FN pueden sufrir anomalías cuando se realizan actualizaciones.

- Para pasar una relación en 1FN a 2FN hay que eliminar las dependencias parciales de la clave primaria. Para ello se eliminan los atributos, que son funcionalmente dependientes, y se ponen en una nueva relación con una copia de su determinante (los atributos de la clave primaria de los que dependen).
- Se crearán dos tablas para eliminar las dependencias funcionales, una de ellas tendrá los atributos que dependen funcionalmente de la clave, y la otra los atributos que forman parte de la clave de la que dependen

• **Ejemplo**: supongamos que tenemos una relación ALUMNO en la que representamos los datos de los alumnos y las notas en cada una de las asignaturas en que está matriculado. La clave es el número de matrícula Cod_Alumno y la asignatura Asignatura.

COD ALUMNO	NOMBRE	APELLIDO	<u>ASIGNATURA</u>	NOTA	CURSO	AULA
1111	PEPE	GARCÍA	LENGUA I	5	1	15
1111	PEPE	GARCÍA	IDIOMAS	5	2	16
2222	MARÍA	SUÁREZ	IDIOMAS	7	2	16
2222	MARÍA	SUÁREZ	CIENCIAS	7	2	14
3333	JUAN	GIL	PLÁSTICA	6	1	18
3333	JUAN	GIL	MATEMÁTICAS I	6	1	12
4444	FRANCISCO	MONTOYA	LENGUA II	4	2	11
4444	FRANCISCO	MONTOYA	MATEMÁTICAS I	6	1	12
4444	FRANCISCO	MONTOYA	CIENCIAS	8	1	14

Tabla ALUMNO para transformarla en 2FN

- Es obvio que todos los atributos no dependen de la clave completa (COD_ALUMNO, ASIGNATURA). En primer lugar, hay que ver las dependencias funcionales de cada uno de los atributos con respecto a los atributos de la clave y el resto de atributos:
 - Nombre y Apellidos, sólo dependen de Cod_Alumno
 - Los atributos Curso y Aula están relacionados con Asignatura, es decir, existe una dependencia entre Asignatura Curso, Asignatura → Aula. Una asignatura pertenece a un único curso y se imparte en un aula (esto me lo dirían en el enunciado).
 - El atributo Nota depende funcionalmente de la clave, pues para que haya una nota tienen que haber una asignatura y un alumno

 Vista las dependencias funcionales llegamos a la siguiente conclusión para que la relación Alumno esté en 2FN necesitamos crear tres relaciones: ALUMNO, ASIGNATURAS

Y NOTAS

COD ALUMNO	NOMBRE	APELLIDO
1111	PEPE	GARCÍA
2222	MARÍA	SUÁREZ
3333	JUAN	GIL
444	FRANCISCO	MONTOYA

Tabla ALUMNO, en 2FN

<u>ASIGNATURA</u>	CURSO	AULA
LENGUA I	1	15
IDIOMAS	2	16
CIENCIAS	2	14
PLÁSTICA	1	18
MATEMÁTICAS I	1	12
LENGUA II	2	11

Tabla ASIGNATURA, en 2FN

COD ALUMNO(FK)	ASIGNATURA(FK)	NOTA
1111	LENGUA I	5
1111	IDIOMAS	5
2222	IDIOMAS	7
2222	CIENCIAS	7
3333	PLÁSTICA	6
3333	MATEMÁTICAS I	6
4444	LENGUA II	4
4444	MATEMÁTICAS I	6
444	CIENCIAS	8

Tabla NOTA, en 2FN

- Para pasar una relación de 2FN a 3FN hay que eliminar las dependencias transitivas. Para ello se eliminan los atributos que dependen transitivamente y se ponen en una nueva relación con una copia de su determinante (el atributo o atributos no clave de los que dependen).

• **Ejemplo**: supongamos que tenemos una relación LIBROS en la que representamos los datos de las editoriales de los mismos

COD_LIBRO	TÍTULO	EDITORIAL	PAÍS
12345	DISEÑO DE BD RELACIONALES	RAMA	ESPAÑA
34562	INSTALACIÓN Y MANTENIMIENTO DE EQUIPOS	MCGRAW-HILL	ESPAÑA
72224	FUNDAMENTOS DE PROGRAMACIÓN	SANTILLANA	ESPAÑA
34522	BASE DE DATOS OO	ADDISON	EEUU

- Tabla LIBROS, para transformar a 3FN
- Veamos las dependencias con respecto a la clave:
 - TÍTULO Y EDITORIAL dependen directamente del código del libro
 - El PAÍS, aunque es parte, también depende del libro, está más ligado a la Editorial a la que pertenece el libro. Por esta razón, la relación libros no está en 3FN. La solución sería:

COD LIBRO	TÍTULO	EDITORIAL (FK)
12345	DISEÑO DE BD RELACIONALES	RAMA
34562	INSTALACIÓN Y MANTENIMIENTO DE EQUIPOS	MCGRAW-HILL
72224	FUNDAMENTOS DE PROGRAMACIÓN	SANTILLANA
34522	BASE DE DATOS OO	ADDISON

<u>EDITORIAL</u>	PAÍS
RAMA	ESPAÑA
MCGRAW-HILL	ESPAÑA
SANTILLANA	ESPAÑA
ADDISON	EEUU

• Tabla Libros, en 3FN

Tabla Editorial en 3FN

• **Ejercicio**: Normalizar hasta la 3FN

DNI	NOMBRE	APELLIDOS	DIRECCIÓN	СР	POBLACIÓN	PROVINCIA
413246-B	JUAN	RAMOS	C/Las Cañas, 59 C/Pilón 12	19005 45589	Guadalajara Caleruela	Guadalajara Toledo
23456-J	PEDRO	PÉREZ	C/Victoria, 3 C/El altozano	28804 10392	Alcalá de Henares Berrocalejo	Madrid Cáceres
34561-B	MARÍA	RODRÍGUEZ	C/Sanz Vázquez,2	19004	Guadalajara	Guadalajara
222346-J	JUAN	CABELLO	C/El ensanche, 3 C/ Los abedules10	28802 10300	Alcalá de Henares Navalmoral	Madrid Cáceres

3.7. OTRAS FORMAS NORMALES

 Existen más formas normales. La forma Boyce-codd (FNBC), es una forma normal estrictamente más sólida que la 3FN pero que en contadas ocasiones genera un modelo distinto a 3FN. También existen 4FN y 5FN, cuyo alcance excede de este curso y cuya aplicación en el mundo real es únicamente teórica.