


| | | | | | | | | |
|---|------------|----------------|--------------|--|--------|-----------|----|--|
|  | RAMA: | Informática | CICLO: | Desenvolvimento de Aplicacions Multiplataforma | | | | |
| | MÓDULO: | Bases de datos | | | | CURSO: | 1º | |
| | PROTOCOLO: | Apuntes clases | EVAL: | 1 | FECHA: | 2023/2024 | | |
| | UNIDAD | | COMPETENCIA: | | | | | |

Tema 3

El Modelo relacional

Índice

| | |
|---|----|
| 1. El modelo relacional | 1 |
| 1.1. Relaciones | 1 |
| 1.2. Atributos y dominios. | 1 |
| 1.3. Claves..... | 2 |
| 1.4. Otros elementos del modelo relacional | 3 |
| 1.5. Propiedades de las relaciones..... | 4 |
| 2. Transformación de un esquema de Entidad-Relación a un esquema relacional | 4 |
| 2.1. Entidades y atributos..... | 5 |
| 2.2. Relaciones | 5 |
| 2.2.1. Relaciones N:M | 6 |
| 2.2.2. Relaciones 1:N..... | 7 |
| 2.2.3. Relaciones 1:1 | 8 |
| 2.2.4. Relaciones terciarias y N-arias | 8 |
| 2.2.5. Relaciones reflexivas | 9 |
| 2.3. Especialización-Generalización | 10 |
| 3. Normalización | 11 |
| 3.1. Dependencias funcionales..... | 11 |
| 3.2. Primera forma normal (1FN) | 13 |
| 3.3. Segunda forma normal (2FN) | 15 |
| 3.4. Tercera forma normal (3FN) | 16 |
| 3.5. Desnormalización | 17 |
| 4. Bibliografía | 17 |

| | | | | | | |
|------------------------------|---------------------|----------------|--------|--|--------|-----------|
| COLEXIO VIVAS S.L. | RAMA: | Informática | CICLO: | Desenvolvimento de Aplicacions Multiplataforma | | |
| | MÓDULO: | Bases de datos | | | | CURSO: 1º |
| | PROTOCOLO: | Apuntes clases | EVAL: | 1 | FECHA: | |
| | UNIDAD COMPETENCIA: | | | | | |

1. El modelo relacional

El modelo relacional es un modelo de datos basado en la teoría de conjuntos ¹ y la lógica de predicados². Fue propuesto por Codd³ a principios de los años 70 y es, actualmente, el paradigma más usado dentro de las bases de datos.

1.1. Relaciones

En este modelo los datos se estructuran de forma lógica en relaciones, se pueden ver como tablas, que contienen los siguientes elementos:

- **Nombre**: es el nombre de la relación, se representa como una cadena de texto sobre la relación.
- **Atributo**: es una propiedad de la relación. Se representa como una columna en la tabla.
- **Tupla**: representa el conjunto de propiedades de cada entidad que participa en la relación. Se representa como una fila de la tabla.
- **Valor**: es el valor concreto para un atributo de una entidad que participen en la relación. Se obtiene de la intersección de una fila con una columna. Puede tomar un valor nulo (null) que representa el desconocimiento o ausencia de valor.

Nombre de la relación → **Coche**

| Matrícula | Modelo | Marca | Color |
|-----------|--------|---------|--------|
| 1111AAA | Focus | Ford | Rojo |
| 2222BBB | Clio | Renault | Blanco |
| 3333CCC | Ibiza | Seat | Negro |
| 4444EEE | Corsa | Opel | Azul |

Tuplas (indicadas por flechas rojas a las filas)

Ejemplos de valores (indicados por flechas azules a los datos)

Atributos (indicados por flechas verdes a las columnas)

1.2. Atributos y dominios.

Como se comento en el punto anterior un atributo es una propiedad que caracteriza una relación.

Cada atributo ha de poseer un dominio. Este representa el conjunto de todos los posibles valores que un atributo puede tomar. Los dominios tienen un nombre con el se puede referenciar, son homogéneos (valores del mismo tipo) e incluyen el valor nulo.

¹ https://es.wikipedia.org/wiki/Teoría_de_conjuntos

² https://es.wikipedia.org/wiki/Lógica_de_primer_orden

³ https://es.wikipedia.org/wiki/Edgar_Frank_Codd

| | | | | | | |
|------------------------------|---------------------|----------------|--------|--|--------|-----------|
| COLEXIO VIVAS S.L. | RAMA: | Informática | CICLO: | Desenvolvemento de Aplicacions Multiplataforma | | |
| | MÓDULO: | Bases de datos | | | | CURSO: 1º |
| | PROTOCOLO: | Apuntes clases | EVAL: | 1 | FECHA: | |
| | UNIDAD COMPETENCIA: | | | | | |

Se pueden definir de dos maneras:

- Extensión: incluyendo todos sus valores. Por ejemplo un dominio que representa los colores de un grupo de coches puede ser: rojo, blanco, negro, azul y rojo. Obliga a que la columna color solo puede tomar uno de esos valores.
- Intensión: indicando la definición de sus valores. Por ejemplo la edad escolar obligatoria de un alumno será: los números enteros comprendidos entre 6 y 16.

Ejemplos de definición de dominios por intención son:

| Atributo | Dominio |
|-------------------|--|
| Matricula | Conjunto de cuatro enteros y tres caracteres |
| Modelo | Conjunto de 12 caracteres |
| Número de puertas | Conjunto de los enteros de una cifra |

1.3. Claves

Un atributo clave es el atributo o conjunto de atributos que permiten identificar de forma unívoca cada tupla (fila) de una relación.

Hay varias clases de claves:

- **Clave candidata.**
Cualquiera de las claves que existen en una relación. Imponen la restricción que no pueden existir dos duplas con el mismo valor de una clave candidata. Por ejemplo en la relación coche pueden existir dos claves candidatas: matrícula y número de bastidor. Esto obliga a que no pueden existir dos coches distintos con la misma matrícula o el mismo número de bastidor.
- **Clave primaria (Primary Key).**
Clave candidata que se ha escogido como identificador de la relación (se recomienda elegir claves primarias cortas). Impone además la restricción que la clave primaria no puede tomar valores nulos.
- **Clave alternativa.**
El resto de las claves candidatas que no son clave primaria.
- **Clave foránea (Foreign Key).**
Es un atributo o un conjunto de atributos que se corresponde con la clave primaria de otra relación.

Por ejemplo, si tenemos las siguientes relaciones:

| Equipo | | | Jugador | | | | |
|----------------|------------------|---------|---------------|--------|----------|-----------|--------|
| <u>CodEqui</u> | Nombre | AñoFun. | <u>CodJug</u> | Nombre | Apellido | Posición | Equipo |
| 1 | Rápido de Bouzas | 1914 | 1 | Veiga | Lorenzo | Medio | 1 |
| 2 | Fabril | 1914 | 2 | Calvo | Pablo | Defensa | 2 |
| 3 | Ponte Ourense | 1977 | 3 | Trillo | Manuel | Delantero | null |
| | | | 4 | Graiño | Xabi | Delantero | 1 |

Clave foránea

| | | | | | | |
|------------------------------|---------------------|----------------|--------|--|--------|-----------|
| COLEXIO VIVAS S.L. | RAMA: | Informática | CICLO: | Desenvolvimento de Aplicacions Multiplataforma | | |
| | MÓDULO: | Bases de datos | | | | CURSO: 1º |
| | PROTOCOLO: | Apuntes clases | EVAL: | 1 | FECHA: | |
| | UNIDAD COMPETENCIA: | | | | | |

Podemos definir una clave foránea entre el campo equipo de la relación jugador y el campo CodEqui de la relación equipo. Esta clave foránea está indicando el equipo de cada jugador.

Las claves foráneas tienen las siguientes características:

- La clave foránea, columna equipo, puede tomar valores nulos que indican información desconocida o que no aplicable a este atributo.
- Todos los valores presentes en la clave foránea tienen que existir en la relación referenciada, en este caso Equipo, a excepción del valor nulo. En la relación jugador los valores no nulos de la columna equipo (en el ejemplo 1 y 2) deben existir en la relación y campo referenciados por la clave foránea. En el ejemplo deben ser valores que existan en el atributo CodEqui de la relación equipo.
- Puede contener valores duplicados. En nuestro ejemplo los jugadores con código 1 y 4 juegan en el mismo equipo (en el rápido de Bouzas).
- Sobre una misma relación se pueden definir más de una clave foránea entre las mismas u otras relaciones.

1.4. Otros elementos del modelo relacional

Además, dentro del modelo relacional, podemos distinguir otros elementos. Tomaremos para los ejemplos la siguiente tabla:

Coches

| <u>Matrícula</u> | Modelo | Marca | Color |
|------------------|-------------|---------|-------|
| 1111AAA | Focus | Ford | Verde |
| 2222BBB | Saxo | Citroën | Azul |
| 3333CCC | SpaceTourer | Citroën | Gris |

- **Grado:** es el número de atributos (columnas) de una relación → 4.
- **Cardinalidad:** es el número de tuplas (filas) de una relación → 3.
- **Cabecera de la relación:** son todos los pares atributo:dominio de la relación.

(**Matrícula**:matrícula, **Modelo**:cadena, **Marca**:cadena, **Color**:cadena)


- **Esquema de la relación:** es la combinación del nombre con la cabecera.

Coches(**Matrícula**:matrícula, **Modelo**:cadena, **Marca**:cadena, **Color**:cadena)

El esquema se suele representar, por comodidad, sin especificar los dominios:

Coches(**Matrícula**, **Modelo**, **Marca**, **Color**)

- **Cuerpo de la relación:** son todas las tuplas que componen la relación.
- **Estado de la relación:** representa toda la relación combinado el esquema y el cuerpo.

| | | | | | | | | |
|---|------------|----------------|--------------|--|--------|--|--------|----|
|  | RAMA: | Informática | CICLO: | Desenvolvemento de Aplicacions Multiplataforma | | | | |
| | MÓDULO: | Bases de datos | | | | | CURSO: | 1º |
| | PROTOCOLO: | Apuntes clases | EVAL: | 1 | FECHA: | | | |
| | UNIDAD | | COMPETENCIA: | | | | | |

Podemos comparar los términos vistos con los usados en otros modelos:

| Modelo Entidad-Relación | Modelo relacional | Tablas |
|---|------------------------------|-----------------------|
| Entidad | Relación | Tabla |
| Ocurrencia | Tupla | Fila |
| Atributo | Atributo | Columna |
| Clave / Identificador | Clave | Clave |
| Número atributos de una entidad | Grado | Número de Columnas |
| Número de ocurrencias en una entidad | Cardinalidad | Número de Filas |

1.5. Propiedades de las relaciones

Las propiedades que una relación debe cumplir y que las diferencia de una tabla son:

- Toda relación debe tener un nombre y este debe ser diferente al resto de las relaciones.
- En una misma relación los nombres de los atributos (columnas) deben ser diferentes entre sí.
- En una relación no pueden existir dos tuplas (filas) iguales.
- Tanto las tuplas como los atributos no están ordenados.
- En la intersección de una tupla con un atributo solo puede existir un único valor (no están permitidos los atributos multivaluados). Es decir la relación es plana.
- Existe la restricción que ningún atributo que forma parte de la clave primaria puede tomar o bien valores nulos o bien el mismo valor en más de una tupla (es decir no pueden existir dos tuplas con el mismo valor para una clave primaria).

2. Transformación de un esquema de Entidad-Relación a un esquema relacional

Una vez generado el esquema conceptual de entidad-relación se debe convertir un modelo lógico: el esquema relacional.

Los pasos de transformación son los siguientes:

| | | | | | | |
|------------------------------|---------------------|----------------|--------|--|--------|-----------|
| COLEXIO VIVAS S.L. | RAMA: | Informática | CICLO: | Desenvolvimento de Aplicacions Multiplataforma | | |
| | MÓDULO: | Bases de datos | | | | CURSO: 1º |
| | PROTOCOLO: | Apuntes clases | EVAL: | 1 | FECHA: | |
| | UNIDAD COMPETENCIA: | | | | | |

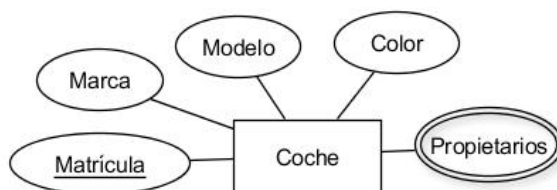
2.1. Entidades y atributos

Las entidades y sus atributos se transforman de la siguiente manera:

- Cada entidad del modelo relacional se transforma en una nueva relación (tabla) con el mismo nombre que la entidad.
- Los atributos de la entidad se convierten en atributos de la relación (columnas de la tabla). Si un atributo es compuesto solo se colocan sus atributos simples. Los atributos derivados no se representan en el modelo relacional.
- La clave de la relación depende de si la entidad:
 - Es fuerte: la clave es la clave primaria de la entidad.
 - Es débil: la clave es la combinación de la clave primaria de la entidad fuerte, de la que depende, y de la entidad débil.
- Los atributos multivaluados se convierten en una nueva tabla siendo su clave primaria una clave compuesta formada por la combinación del atributo clave de la entidad, como clave foránea (FK) más el propio atributo multivaluado.

EJEMPLO 1

Según lo anterior la entidad coche se convierte en el siguiente esquema:



Coche (Matrícula, Marca, Modelo, Color)

PropietariosCoche (Matrícula(Fk), Propietario)

2.2. Relaciones

Para transformar una relación se ha de tener en cuenta las cardinalidades máximas de ambos lados.

Por ejemplo:

- $(0,1) - (0,N) \rightarrow 1:N$
- $(1,1) - (0,1) \rightarrow 1:1$
- $(1,N) - (N,M) \rightarrow N:M$

Combinándose para formar las siguientes reglas de transformación.

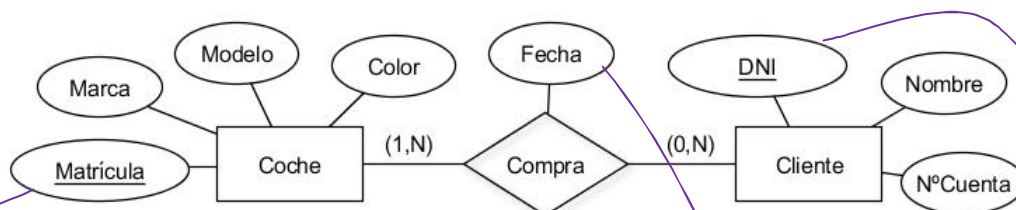
| | | | | | | |
|-------------------------------------|---------------------|----------------|--------|--|--------|-----------|
| COLEXIO VIVAS S.L. | RAMA: | Informática | CICLO: | Desenvolvemento de Aplicacions Multiplataforma | | |
| | MÓDULO: | Bases de datos | | | | CURSO: 1º |
| | PROTOCOLO: | Apuntes clases | EVAL: | 1 | FECHA: | |
| | UNIDAD COMPETENCIA: | | | | | |

2.2.1. Relaciones N:M

Las relaciones N:M se transforman de la siguiente manera:

- Se crea una nueva tabla cuyo nombre será el nombre de la relación en el modelo entidad-relación.
- Su clave estará formada por la unión de las claves de las entidades que participan en la relación. Estas claves como referencian a otras tablas serán claves foráneas: FK (Foreing key).
- Los atributos de la relación, si los hubiera, se añaden a la tabla creada.

EJEMPLO 2



- Siguiendo el punto 2.1 convertimos cada entidad en una tabla.

Coche (Matrícula, Marca, Modelo, Color)

Cliente (DNI, Nombre, NºCuenta)

- Según el punto 2.2.1 creamos una nueva tabla.

Compra (Matricula(Fk), DNICliente(Fk), FechaCompra)

El atributo Matricula de la tabla compra hace referencia a la matricula de un coche presente en la tabla coche (por lo que es clave foránea) mientras que es atributo DNICliente hace referencia a la tabla Cliente (también es clave foránea). Podemos ver que en las claves foráneas no es obligatorio conservar el nombre del atributo referenciado. A esta tabla se añade el atributo presente en la relación: Fecha.

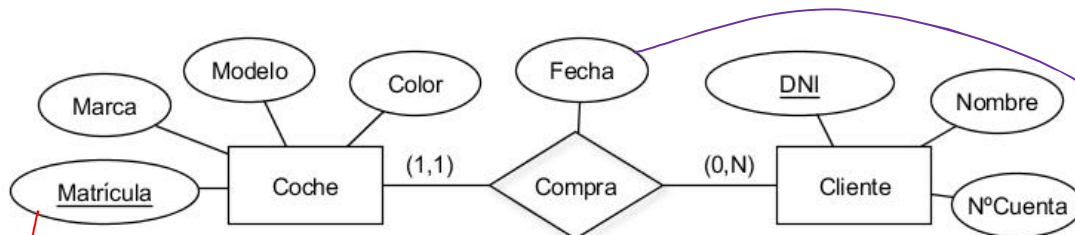
| | | | | | | |
|------------------------------|---------------------|----------------|--------|--|--------|-----------|
| COLEXIO VIVAS S.L. | RAMA: | Informática | CICLO: | Desenvolvimento de Aplicacions Multiplataforma | | |
| | MÓDULO: | Bases de datos | | | | CURSO: 1º |
| | PROTOCOLO: | Apuntes clases | EVAL: | 1 | FECHA: | |
| | UNIDAD COMPETENCIA: | | | | | |

2.2.2. Relaciones 1:N

Para las relaciones 1:N tenemos dos opciones:

- Si el lado con cardinalidad 1 tiene como cardinalidad (1,1) se propaga, como clave foránea, la clave del lado 1 junto con los atributos de la relación, si los hubiera, al lado N. La relación, en este caso compra, desaparece.

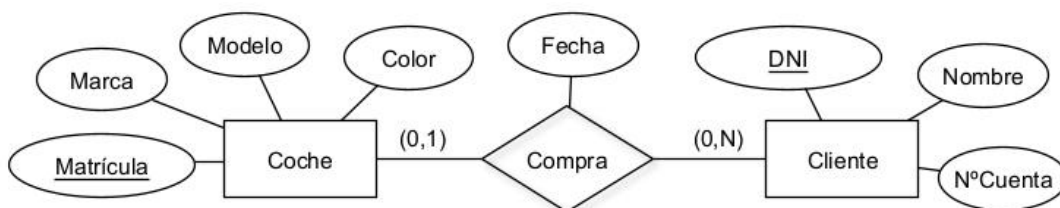
EJEMPLO 3



- Siguiendo el punto 2.1 convertimos cada entidad en una tabla.
 Coche (**Matricula**, Marca, Modelo, Color)
 Cliente (~~DNI~~, Nombre, NºCuenta)
- Siguiendo el punto 2.3 propagamos la clave y los atributos del lado 1 (coche) al lado N (Cliente) por lo que la relación Cliente quedaría.
 Cliente (**DNI**, Nombre, NºCuenta, **Matricula(Fk)**, Fecha)

- Si el lado con cardinalidad 1 tiene como cardinalidad mínima 0 (0,1) se realiza como si fuese una N:M pero siendo la clave primaria únicamente el atributo clave del lado N (de la relación 1:N).

EJEMPLO 4



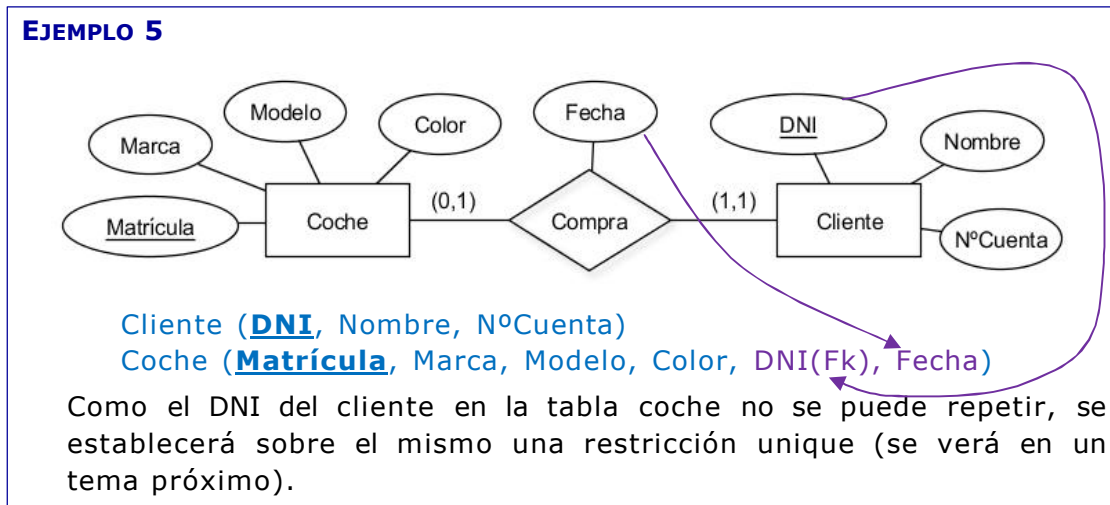
- Convertimos cada entidad en una tabla.
 Coche (**Matricula**, Marca, Modelo, Color)
 Cliente (**DNI**, Nombre, NºCuenta)
- Creamos una tabla nueva, como en una relación N:M, en la que solo es clave primaria el atributo clave que viene del lado 1.
 Compra (**DNICliente(Fk)**, Matricula(Fk), FechaCompra)

| | | | | | | |
|-------------------------------------|---------------------|----------------|--------|---|--------|-----------|
| COLEXIO VIVAS S.L. | RAMA: | Informática | CICLO: | Desenvolvemiento de Aplicacions Multiplataforma | | |
| | MÓDULO: | Bases de datos | | | | CURSO: 1º |
| | PROTOCOLO: | Apuntes clases | EVAL: | 1 | FECHA: | |
| | UNIDAD COMPETENCIA: | | | | | |

2.2.3. Relaciones 1:1

En las relaciones 1:1 tenemos las siguientes opciones:

- Si ambas entidades participan en la relación con cardinalidades (0,1) se procede como en las relaciones N:M.
- Si en la relación tenemos la cardinalidad (0,1) en una entidad y (1,1) en la otra se propaga la clave principal, como clave foránea, del lado (1,1) al lado (0,1). Además si la relación tiene atributos estos también se propagan. La relación desaparece.



- Si las dos entidades participan con una cardinalidad (1,1) tenemos dos opciones:
 - Se propaga, arbitrariamente, la clave de una entidad (como foránea) y los atributos de la relación a la otra entidad. La relación desaparece.
 - Unificar todos los atributos de las entidades y de la relación en una única tabla eligiendo una de las claves primarias de las entidades como clave primaria de la tabla. Una de las entidades y la relación desaparece.

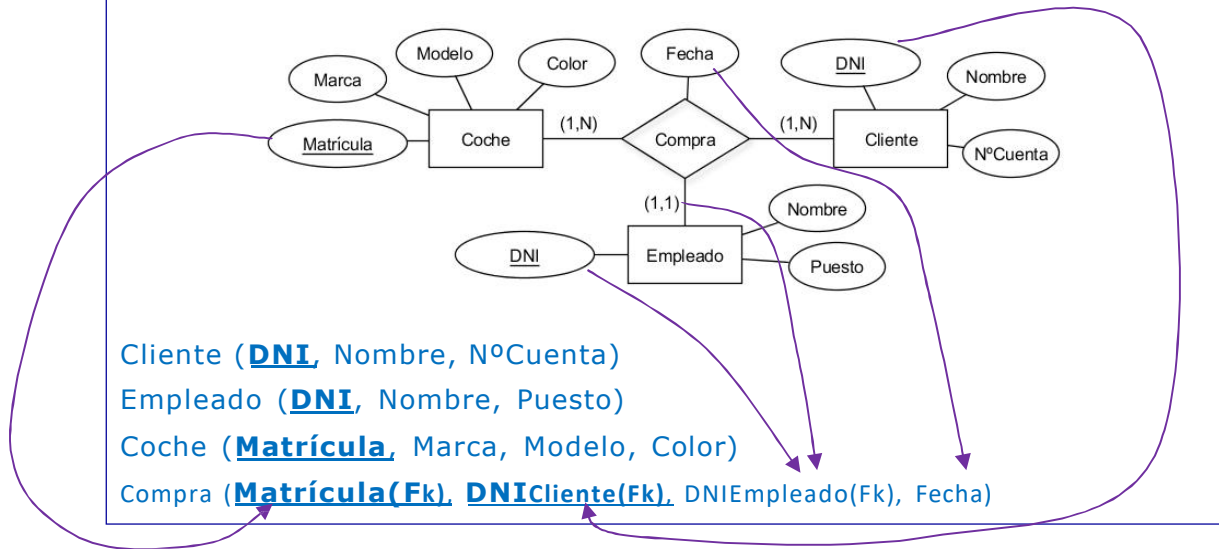
2.2.4. Relaciones terciarias y N-arias

Para transformar estas relaciones se procede de la siguiente manera:

- Cada entidad se transforma en una tabla nueva.
- Se crea una nueva tabla con la relación donde aparecerán tanto las claves primarias de cada entidad, participando como claves foráneas, como los atributos de la relación.
- La clave se forma con las claves de las entidades que participen en la relación con una cardinalidad máxima M.

| | | | | | | |
|------------------------------|---------------------|----------------|--------|--|--------|-----------|
| COLEXIO VIVAS S.L. | RAMA: | Informática | CICLO: | Desenvolvimento de Aplicacions Multiplataforma | | |
| | MÓDULO: | Bases de datos | | | | CURSO: 1º |
| | PROTOCOLO: | Apuntes clases | EVAL: | 1 | FECHA: | |
| | UNIDAD COMPETENCIA: | | | | | |

EJEMPLO 6

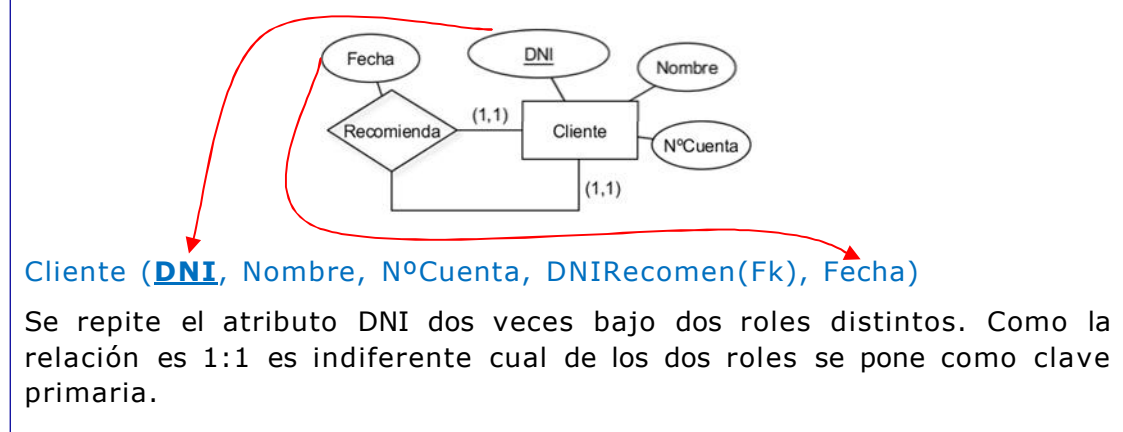


2.2.5. Relaciones reflexivas

Para transformar las relaciones reflexivas tenemos las siguientes tres opciones dependiendo de la cardinalidad de la relación:

- 1:1 → se transforma la relación añadiéndole a la entidad dos veces el atributo clave primaria, una como clave primaria y otra como clave foránea de sí misma.

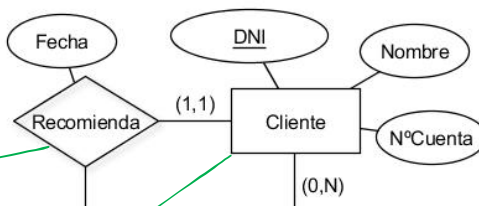
EJEMPLO 7



- 1:N → podemos tener dos casos:
 - Si la entidad N es obligatoria (la parte min es distinta de cero) se transforma de la misma manera que en el caso anterior.
 - Si entidad N no es obligatoria (la parte min es cero), se crea una nueva tabla con los atributos de la clave primaria de la entidad, como clave foránea, además de los atributos de la relación. La clave será el atributo que participe en la relación en la parte N.

| | | | | | |
|---------------------|----------------|--------|--|--------|-----------|
| RAMA: | Informática | CICLO: | Desenvolvimento de Aplicacions Multiplataforma | | |
| MÓDULO: | Bases de datos | | | | CURSO: 1º |
| PROTOCOLO: | Apuntes clases | EVAL: | 1 | FECHA: | |
| UNIDAD COMPETENCIA: | | | | | |

EJEMPLO 8



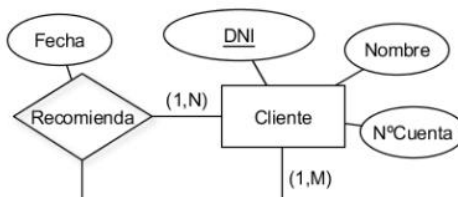
En la tabla recomienda se introduce el DNI bajo los dos roles, pero solo el de la parte N es clave primaria. Los dos son claves foráneas porque referencian al DNI de la tabla cliente.

Cliente (DNI, Nombre, NºCuenta)

Recomienda (DNIRecomendado(Fk), DNIRecomendador(Fk), fecha)

- N:M → se realizan exactamente las mismas acciones que en el caso de dos relaciones binarias con cardinalidad N:M con la salvedad que en este caso la clave primaria de la relación es la combinación dos veces la clave primaria de la entidad bajo dos roles diferentes.

EJEMPLO 9



Cliente (DNI, Nombre, NºCuenta)

Recomienda (DNIRecomendado(Fk), DNIRecomendador(Fk), fecha)

2.3. Especialización-Generalización

Para representar las especializaciones-generalizaciones se deberá crear una relación 1:1 entre la superclase y las subclases siendo esta (0,1) o (1,1) si es solapada y (0,1) si es de disyunción.

EJEMPLO 10


Por un lado se transforma la superclase y luego las subclases cuya clave es la clave de la superclase (como clave foránea).

Coche (Matrícula, Marca, Modelo, Color)

Nuevos (Matrícula(Fk), Garantía)

SegundaMano(Matrícula(Fk), Antigüedad)



| | | | | | | | | |
|---|---------------------|----------------|--------|--|--------|--|--------|----|
|  | RAMA: | Informática | CICLO: | Desenvolvimento de Aplicacions Multiplataforma | | | | |
| | MÓDULO: | Bases de datos | | | | | CURSO: | 1º |
| | PROTOCOLO: | Apuntes clases | EVAL: | 1 | FECHA: | | | |
| | UNIDAD COMPETENCIA: | | | | | | | |

3. Normalización

El proceso de normalización tiene como objetivo la eliminación de dependencias, no buscadas, entre atributos que pueden provocar anomalías en las inserciones, actualizaciones y borrados de datos en la base de datos.

Este proceso consta de una serie de pasos llamados formas normales. Las formas normales más habituales son: 1FN (primera forma normal), 2FN (segunda forma normal), 3FN (tercera forma normal), FNBC (forma normal de Boyce-Codd), 4FN (cuarta forma normal) y 5FN (quinta forma normal).

Estas formas normales son inclusivas, para que una relación este en 2FN tiene que estar previamente en 1FN, para que este en 3FN tiene que estar en 2FN, y por lo tanto en 1FN, y así sucesivamente.

No es obligatorio normalizar hasta la forma normal más alta (las situaciones para poder aplicar la 4FN y 5FN son muy particulares). Por ese motivo solo se verá como normalizar hasta la 3FN ya que es suficiente para la mayoría de las situaciones.


3.1. Dependencias funcionales

Una dependencia funcional es una relación conceptual entre atributos que pertenecen a una misma relación (tabla). No existen reglas para determinar estas dependencias puesto que es el usuario el que, en base al significado que le otorga a los atributos, establece su posible relación.

Si A y B son dos atributos, o un conjunto de ellos, cualquiera de una relación, se puede decir que $A \rightarrow B$ **si el mismo valor de A, llamado determinante, determina siempre el mismo valor en B**. Esta relación de dependencia tiene las siguientes interpretaciones según sea su sentido:

- De izquierda a derecha: A determina B.
- De derecha a izquierda: B es funcionalmente dependiente de A.

| Relación de dependencia | Significado |
|---|--|
| DNI \rightarrow NombrePersona | Con el DNI podemos determinar el nombre de una persona. |
| CodProducto \rightarrow { Marca, Modelo } | La marca y el modelo son determinados por su CodProducto. |
| { CodConductor, NumAutobús } \rightarrow KilometrosConducidos | Los atributos código del conductor y número de autobús permiten determinar cuántos kilómetros ha conducido cada conductor cada autobús. Solo con CodConductor no obtendríamos el valor buscado ya que solo podríamos conseguir la suma de los kilómetros conducidos por ese conductor en todos los autobuses. Tampoco numAutobus lo permite ya que obtendríamos la suma de los kilómetros totales que ha sido conducido un autobús. Por lo que para obtener el valor buscado se necesitan obligatoriamente los dos atributos. |

| | | | | | | | | |
|---|------------|----------------|--------------|--|---|--------|--------|--|
|  | RAMA: | Informática | CICLO: | Desenvolvemento de Aplicacions Multiplataforma | | | | |
| | MÓDULO: | Bases de datos | | | | CURSO: | 1º | |
| | PROTOCOLO: | Apuntes clases | | EVAL: | 1 | | FECHA: | |
| | UNIDAD | | COMPETENCIA: | | | | | |

Existen una serie de relaciones, que se denominan triviales, que son cumplidas por todas las relaciones:

- $A \rightarrow A$: A se incluye a sí misma. Por ejemplo con el DNI determino el DNI.
- $A \rightarrow B$: si B es un subconjunto de A. Por ejemplo con el DNI y el nombre determino el DNI.

Podemos encontrar los siguientes tipos de dependencias:

➤ Dependencia funcional completa

Un atributo, o un conjunto de ellos, B es completamente dependiente de un conjunto de atributos A si depende de A pero **no depende** funcionalmente de **ningún atributo** que forme parte de A.

| | |
|---|---|
| A | B |
| ----- | |
| { CodConductor, NumAutobús } → KilometrosConducidos | |


KilometrosConducidos depende completamente de CodConductor y NumAutobús ya que depende funcionalmente de ellos (de los dos de forma conjunta) y no depende de ninguno de ellos por separado. Solo con CodConductor o solo con NumAutobús no puedo determinar KilometrosConducidos.

➤ Dependencia funcional parcial

Un atributo, o un conjunto de ellos, B tiene una dependencia parcial de un conjunto de atributos A si depende de A pero **también depende** funcionalmente de **algún atributo** que forme parte de A.

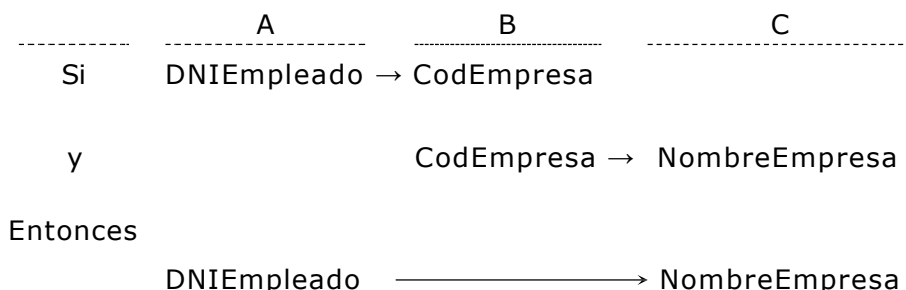
| | |
|---|---|
| A | B |
| ----- | |
| { CodConductor, NumAutobús } → { NombreConductor, FechaNacimiento } | |
| CodConductor → { NombreConductor, FechaNacimiento } | |

NombreConductor y FechaNacimiento dependen parcialmente de CodConductor y NumAutobús ya que depende funcionalmente de ellos (de los dos de forma conjunta) y además depende de una parte de ellos: CodConductor. Para determina el conjunto de atributos B solo hace falta CodConductor (un subconjunto de A). NumAutobús no aporta ninguna información relevante a la relación.

| | | | | | | |
|---|---------------------|----------------|--------|--|--------|-----------|
|  | RAMA: | Informática | CICLO: | Desenvolvimento de Aplicacions Multiplataforma | | |
| | MÓDULO: | Bases de datos | | | | CURSO: 1º |
| | PROTOCOLO: | Apuntes clases | EVAL: | 1 | FECHA: | |
| | UNIDAD COMPETENCIA: | | | | | |

➤ Dependencia transitiva

Un atributo, o conjunto de atributos, C depende transitivamente de A si C depende funcionalmente de B y B depende funcionalmente de A.



Si con DNI del empleado se puede determinar el código de la empresa y con este se puede determinar el nombre de la empresa entonces con el DNI del empleado se puede determinar, de forma transitiva, el nombre de la empresa.

3.2. Primera forma normal (1FN)

La 1FN es una regla inherente al modelo relacional, ver punto 1.5, y es la única regla formal obligatoria que toda relación debe cumplir.

Una relación esta en 1FN si:

- El valor de cada atributo es atómico, es decir la intersección de una fila con una columna tiene un único valor.
- No tiene grupos repetitivos. Imaginemos el caso de que se desea almacenar los teléfonos de un cliente. Se podría decidir que cada cliente tiene tres teléfonos por lo que creamos la relación teléfonos(nombre, apellido, telf1, telf2, telf3). Estos tres campos forman un grupo repetitivo ya que representan un mismo atributo teléfono. Esta solución presenta los siguientes problemas:
 - Si un cliente no tiene tres teléfonos aparecen atributos con valor nulo.
 - Se impone la restricción que como mucho un cliente puede tener tres teléfonos, ¿qué pasa si un cliente tiene cuatro teléfonos?

Para pasar una relación a 1FN se tiene que eliminar los valores múltiples y los grupos repetitivos. Existen dos formas de llevarlo a cabo:

- Crear una fila nueva para cada valor del atributo repetido y el resto de atributos se rellenan con sus correspondientes valores simples. **La clave primaria se establece con la combinación de la clave primaria original más el atributo que posee varios valores**. Esto inserta redundancias en la relación que se eliminarán durante la aplicación del resto de formas normales.

| | | | | | | |
|------------------------------|---------------------|----------------|--------|---|--------|-----------|
| COLEXIO VIVAS S.L. | RAMA: | Informática | CICLO: | Desenvolvemiento de Aplicacions Multiplataforma | | |
| | MÓDULO: | Bases de datos | | | | CURSO: 1º |
| | PROTOCOLO: | Apuntes clases | EVAL: | 1 | FECHA: | |
| | UNIDAD COMPETENCIA: | | | | | |

EJEMPLO 11

La tabla siguiente representa los clientes de una tienda, en la cual, los clientes pueden poseer más de un teléfono, lo que provoca valores múltiples para el atributo teléfono, por lo que no está en 1FN.

Cientes

| <u>CodCliente</u> | Nombre | Apellido | Teléfono |
|-------------------|--------|----------|-------------------------------------|
| 1 | Juan | Álvarez | 555000001 555000002 |
| 2 | Luis | Justo | 555111111 |
| 3 | Marcos | Campos | 555222221 555222222 555222223 |

Para convertirla a 1FN se eliminan los valores repetidos creando nueva filas con ellos y estableciendo como nueva clave la combinación de la clave existente (CodCliente) y el atributo con valores múltiples (Teléfono).

Cientes

| <u>CodCliente</u> | Nombre | Apellido | <u>Teléfono</u> |
|-------------------|--------|----------|-----------------|
| 1 | Juan | Álvarez | 555000001 |
| 1 | Juan | Álvarez | 555000002 |
| 2 | Luis | Justo | 555111111 |
| 3 | Marcos | Campos | 555222221 |
| 3 | Marcos | Campos | 555222222 |
| 3 | Marcos | Campos | 555222223 |

- Mover el atributo con múltiples valores a una nueva tabla cuya clave primaria será la combinación de la clave original de la relación más el propio atributo.

EJEMPLO 12

Por lo que la relación anterior se convertirían en las dos relaciones siguientes:

Cientes

| <u>CodCliente</u> | Nombre | Apellido |
|-------------------|--------|----------|
| 1 | Juan | Álvarez |
| 2 | Luis | Justo |
| 3 | Marcos | Campos |

Teléfonos

| <u>CodCliente(Fk)</u> | <u>Teléfono</u> |
|-----------------------|-----------------|
| 1 | 555000001 |
| 1 | 555000002 |
| 2 | 555111111 |
| 3 | 555222221 |
| 3 | 555222222 |
| 3 | 555222223 |

| | | | | | | |
|------------------------------|---------------------|----------------|--------|--|--------|-----------|
| COLEXIO VIVAS S.L. | RAMA: | Informática | CICLO: | Desenvolvimento de Aplicacions Multiplataforma | | |
| | MÓDULO: | Bases de datos | | | | CURSO: 1º |
| | PROTOCOLO: | Apuntes clases | EVAL: | 1 | FECHA: | |
| | UNIDAD COMPETENCIA: | | | | | |

3.3. Segunda forma normal (2FN)

Una relación está en 2FN si esta en 1FN y todo atributo que no pertenece a la clave primaria depende funcionalmente de forma completa de esta.

Es decir, todo atributo que no forme parte de la clave primaria tiene que depender de toda la clave primaria, ya que esta puede estar formada por varios atributos, y no solo de algún atributo de los que la formen. Por este motivo si **una relación está en 1FN y su clave está formada por un único atributo está automáticamente en 2FN**.

Una relación en 1FN se convierte a 2FN moviendo, eliminándolos de la relación original, los atributos que dependen de forma parcial de la clave primaria a una nueva relación cuya clave será el atributo de la clave primaria del que dependen. La relación original, menos los atributos eliminados, pertenece también a la 2FN.

EJEMPLO 13

Imaginemos que tenemos la tabla ventas que representan los viajes que han sido vendidos a los clientes. El mismo viaje, dependiendo de la fecha de compra, puede tener precios diferentes.

Viajes

| <u>CodCliente</u> | Nombre | Apellido | <u>CodViaje</u> | Destino | Precio |
|-------------------|--------|----------|-----------------|------------|--------|
| 1 | Juan | Álvarez | 1 | Nueva York | 800 |
| 1 | Juan | Álvarez | 2 | Paris | 700 |
| 2 | Luis | Justo | 2 | Paris | 550 |
| 2 | Luis | Justo | 3 | Ibiza | 400 |
| 3 | Marcos | Campos | 1 | Nueva York | 900 |
| 3 | Marcos | Campos | 4 | Berlín | 650 |
| 3 | Marcos | Campos | 2 | Paris | 625 |

Se puede observar que no todos los atributos dependen de los dos atributos que forman la clave primaria (recordemos que $A \rightarrow B$ si el mismo valor de A determina siempre el mismo valor en B):

- Nombre y apellido solo dependen de parte de la clave primaria: el código del cliente. Se puede comprobar que con un mismo valor de CodViaje determino nombres y apellidos distintos.
- El destino también depende de parte de la clave primaria, pero en este caso del código de viaje. No depende de CodCliente puesto que con un mismo valor de CodCliente determino destinos distintos.
- El resto de atributos, en nuestro caso precio, depende de forma completa de la clave primaria (de todos sus atributos). Con el código de cliente no se cumple la dependencia CodCliente \rightarrow Precio ya que para el código de cliente 1 tenemos dos filas con precio distinto. Con el código de viaje tampoco se cumple CodViaje \rightarrow Precio ya que el código de viaje 2 tiene dos precios distintos.

| | | | | | | |
|------------------------------|---------------------|----------------|--------|--|--------|-----------|
| COLEXIO VIVAS S.L. | RAMA: | Informática | CICLO: | Desenvolvimento de Aplicacions Multiplataforma | | |
| | MÓDULO: | Bases de datos | | | | CURSO: 1º |
| | PROTOCOLO: | Apuntes clases | EVAL: | 1 | FECHA: | |
| | UNIDAD COMPETENCIA: | | | | | |

Según las dependencias funcionales indicadas la relación anterior se transformaría en tres relaciones: Ventas, viajes y clientes.

Clientes

| <u>CodCliente</u> | Nombre | Apellido |
|-------------------|--------|----------|
| 1 | Juan | Álvarez |
| 2 | Luis | Justo |
| 3 | Marcos | Campos |

Destinos

| <u>CodViaje</u> | Destino |
|-----------------|------------|
| 1 | Nueva York |
| 2 | Paris |
| 3 | Ibiza |
| 4 | Berlín |

Viajes

| <u>CodCliente(Fk)</u> | <u>CodViaje(Fk)</u> | Precio |
|-----------------------|---------------------|--------|
| 1 | 1 | 800 |
| 1 | 2 | 700 |
| 2 | 2 | 550 |
| 2 | 3 | 400 |
| 3 | 1 | 900 |
| 3 | 4 | 650 |
| 3 | 2 | 625 |

- Como nombre y apellido solo dependen del código de cliente se mueven a una tabla cuya clave será código de cliente que es el atributo, de la clave primaria, del que dependen.
- Con el destino se hace lo mismo, se mueve a una tabla nueva cuya clave será el código del viaje, el atributo de la clave primaria del que depende.
- El único cambio en la tabla ventas son los atributos que se han eliminado, en nuestro caso el nombre, apellido y el destino.

3.4. Tercera forma normal (3FN)

Una relación está en 3FN si esta en 2FN (y por lo tanto esta en 1FN) y cada atributo que no pertenece a la clave primaria no depende de forma transitiva de esta. Es decir no existe ningún atributo intermedio que dependa de la clave primaria y que de él dependan otros atributos.

La conversión de una en 2FN a 3FN es similar a la vista en el punto anterior, pero en vez eliminar dependencias parciales se eliminan dependencias transitivas. Todos los atributos que dependan de un atributo A que no esté en la clave primaria se mueven a otra relación cuya clave será el atributo A.

EJEMPLO 14

Supongamos que tenemos una relación con los destinos de los viajes y sus barrios: Vemos que está en 1FN ya que no hay valores múltiples y como la clave tiene un único atributo está automáticamente en 2FN.

Viajes

| <u>CodViaje</u> | Destino | <u>CodBarrio</u> | Nombre |
|-----------------|------------|------------------|------------|
| 1 | Nueva York | 1 | Bronx |
| 2 | Paris | 3 | Montmartre |
| 3 | Nueva York | 2 | Brooklyn |
| 4 | Nueva York | 2 | Brooklyn |
| 5 | Paris | 5 | Le Marais |

| | | | | | | |
|-------------------------------------|---------------------|----------------|--------|--|--------|-----------|
| COLEXIO VIVAS S.L. | RAMA: | Informática | CICLO: | Desenvolvimento de Aplicacions Multiplataforma | | |
| | MÓDULO: | Bases de datos | | | | CURSO: 1º |
| | PROTOCOLO: | Apuntes clases | EVAL: | 1 | FECHA: | |
| | UNIDAD COMPETENCIA: | | | | | |

Para transformarla en 3FN tenemos que eliminar las dependencias transitivas. Se puede ver que aunque todos los atributos dependen de la clave primaria el nombre del barrio tiene una relación más fuerte con el código del barrio que con el código del viaje. Por lo que se procede a mover este atributo a una relación nueva.

Viajes

| <u>CodViaje</u> | Destino | CodBarrio |
|-----------------|------------|-----------|
| 1 | Nueva York | 1 |
| 2 | Paris | 3 |
| 3 | Nueva York | 2 |
| 4 | Nueva York | 2 |
| 5 | Paris | 5 |

Barrios

| <u>CodBarrio</u> | Nombre |
|------------------|------------|
| 1 | Bronx |
| 2 | Brooklyn |
| 3 | Montmartre |
| 5 | Le Marais |

3.5. Desnormalización

En ciertas ocasiones la normalización puede originar una alta fragmentación de los datos debido a la gran cantidad de relaciones generadas lo que provoca un aumento de la complejidad y un mayor coste computacional en la ejecución de consultas.

En estos casos el diseñador de la base de datos puede optar por volver a una forma normal anterior, lo que genera la aparición de redundancia en los datos, que debe estar controlada, en aras de reducir la complejidad y/o aumentar la eficiencia de las consultas.

4. Bibliografía

1. [Teoría de conjuntos](#)
2. [Lógica de primer orden](#)
3. [Edgar Frank Codd](#)