

Gestión de Datos para Robótica

T1d - Modelo Relacional

Álvaro Vázquez Álvarez
Departamento de Electrónica e Computación

✉ alvaro.vazquez@usc.es

📍 Pabellón III - Despacho 4

Curso 2023-2024

Tabla de contenidos

- Orígenes del Modelo Relacional
 - Concepto de Relación
 - Estructura de Datos Relacional
 - Restricciones del Modelo Relacional
- Conversión del Modelo ER a Modelo Relacional
- Bibliografía

Orígenes del Modelo Relacional

Propuesto por Codd en 1970

Los **objetivos** que perseguía con su nuevo modelo eran:

- **Independencia lógica**
- **Independencia física**
- **Flexibilidad**
- **Uniformidad**
- **Sencillez**

Se basa en dos ramas de las matemáticas:

- Teoría de conjuntos
- Lógica de predicados de primer orden

El tener una **base matemática** lo convierte en un modelo **seguro, robusto, fiable y predecible**.

Modelo de Datos: conjunto de **conceptos** que sirven para **describir**, en distintos niveles de abstracción, la **estructura** (esquema) de una BD.

Es decir, **los tipos de datos, relaciones y restricciones** que han de cumplirse, así como las **operaciones** para consultar y actualizar los datos.



Orígenes del Modelo Relacional

Concepto de Relación

- El **modelo relacional** define la **relación** como la **estructura** para **almacenar los datos de la BD**.
- Una **relación** se **parece** a una **tabla** (aunque no es lo mismo)
- La **BD** está formada por un **conjunto** de **relaciones**.
- Si vemos una relación como una tabla donde:
 - **Filas:** representan hechos que se corresponden con una entidad del mundo real.
 - **Nombre de la tabla:** define de manera genérica los hechos almacenados en la tabla
 - **Columnas:** permiten identificar el significado de los valores que están en cada fila.

Una fila es una **tupla**, el nombre de una columna es un **atributo** y la tabla se denomina **relación**



Orígenes del Modelo Relacional

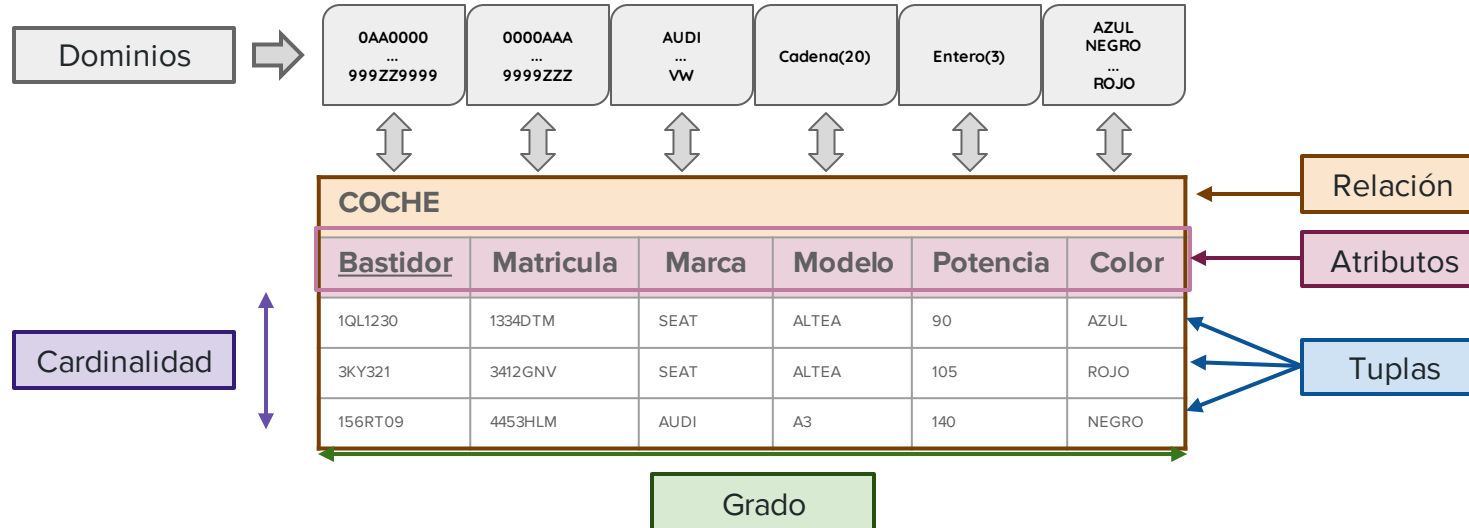
Concepto de Relación

- Una **relación** almacena determinados atributos de entidades reales del mismo tipo
 - Ejemplo: la relación Alumnos almacena los atributos DNI, Nombre y Titulación de todos los alumnos.
- Un **esquema de relación** R , denotado por $R(A_1, A_2, A_3, \dots, A_n)$ está compuesto por un nombre de relación R y una lista de atributos $(A_1, A_2, A_3, \dots, A_n)$.
 - Ejemplo: *Alumnos (DNI, Nombre, Titulación)*
- Un **atributo** admite valores de un determinado dominio (**conjunto** de **valores atómicos**)
 - Ejemplo: DNI \rightarrow números de 8 cifras seguidos de una letra.
- Se denota **dominio** de un atributo A por $dom(A)$.
 - Ejemplo: $dom(titulación) = \{GR, GP, GIC, GIFMA, GIAA\}$
- El **grado** de una relación determina el **número** de **atributos** que tiene la relación.
 - Ejemplo: el grado de la relación Alumnos anterior es tres.
- La **cardinalidad** de la relación indica el **número** de **tuplas** que tiene la relación.

Orígenes del Modelo Relacional

Concepto de Relación

- Ejemplo:

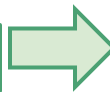


Orígenes del Modelo Relacional

Concepto de Relación

- Definición formal:
 - Una relación r del esquema de relación $R(A_1, A_2, A_3, \dots, A_n)$, denotado por $r(R)$, es un **conjunto** de tuplas $r = \{t_1, t_2, \dots, t_m\}$
 - Cada tupla es una lista ordenada de n valores $t_i = \{v_1, v_2, \dots, v_n\}$ donde cada v_j es un elemento del dominio $dom(A_j)$ o bien es nulo.
 - Una relación $r(R)$ es un subconjunto del producto cartesiano de los dominios de los atributos.
 - $r(R) \subseteq dom(A_1) \times dom(A_2) \times \dots \times dom(A_n)$
- La BD es una **colección** de **relaciones**
 - Ejemplo: Alumnos (DNI, Nombre, Titulación), Asignatura (Nombre, Curso, Cuatrimestre), Profesor (DNI, Nombre, Departamento).

¿Una relación puede tener filas repetidas?



NO, porque es un conjunto (conjunto no admite duplicados)

¿Hay orden entre las tuplas de una relación?



NO, porque no hay orden en los elementos de un conjunto

Orígenes del Modelo Relacional

Estructura de Datos Relacional

- Sea R una relación $R(A_1:D_1, A_2:D_2, \dots, A_n:D_n)$
 - A = Atributo
 - D = Dominio
- **Superclave:** subconjunto de atributos S_k de R , tal que cumplen la restricción de **unicidad**.
 - **Unicidad** → NO existen dos tuplas (t_1, t_2) con la misma combinación de valores para S_k

$$t_1[S_k] \neq t_2[S_k]$$

- **Clave:** es una superclave K de R (conjunto de atributos de R) que cumple la restricción de **irreductibilidad**.
 - **Irreductibilidad** → ningún subconjunto de K cumple la propiedad de unicidad.

Superclave: conjunto de atributos que asegura que las **tuplas** son **únicas**



Clave: no existen atributos redundantes (todos son necesarios para asegurar la unicidad).







Orígenes del Modelo Relacional

Estructura de Datos Relacional

- Ejemplo: dada la relación Coche(Bastidor, Matricula, Marca, Modelo, Potencia, Color)

COCHE					
<u>Bastidor</u>	Matricula	Marca	Modelo	Potencia	Color
1QL1230	1334DTM	SEAT	ALTEA	90	AZUL
3KY321	3412GNV	SEAT	ALTEA	105	ROJO
156RT09	4453HLM	AUDI	A3	140	NEGRO

- Cual de los siguientes conjuntos de atributos son superclave (S_K) y/o clave (K)

	S_K	K	
{Bastidor, Marca, Modelo}	✓	✗	 {Bastidor, Marca, Modelo} $\neq S_K \neq K$
{Marca, Potencia}	✗	✗	 {Marca, Potencia} $\neq S_K \neq K$
{Bastidor, Matricula}	✓	✗	 {Bastidor, Matricula} = S_K {Bastidor, Matricula} = $S_K \rightarrow$ redundantes $\neq K$
{Bastidor}	✓	✓	 {Bastidor} $\neq S_K = K$

Orígenes del Modelo Relacional

Estructura de Datos Relacional

- 5 tipos de claves:
 - **Clave candidata:** cada una de las claves de una relación.
 - **Clave compuesta:** clave candidata compuesta por más de un atributo
 - **Clave primaria (Primary Key):** clave candidata elegida como clave de la relación.
 - Ningún valor de la clave primaria debe ser nulo → **restricción de integridad de entidad**
 - **Clave alternativa:** clave candidata que no ha sido elegida como clave primaria.
 - **Clave foránea (Foreign Key):** clave cuyos valores hacen referencia a los valores de una clave primaria en otra relación.
 - La FK debe tener el mismo dominio que la PK → **restricción de integridad referencial**
 - El valor de una tupla de FK debe tener el mismo valor en una tupla PK o bien ser nulo.



- Una FK puede referenciar a la PK de la misma relación (no es lo habitual)

Orígenes del Modelo Relacional

Estructura de Datos Relacional

- Valor nulo
 - En el mundo real existe información:
 - **Desconocida** en un determinado momento
 - N° de matrícula en el momento de la compra de un coche (si está inmatriculado).
 - **Ausente** porque no se dispone del mismo
 - Número de teléfono de una persona que no tiene.
 - **No aplicable** a ciertos atributos
 - Sueldo a una persona parada.
 - Para representar esa información los SGBD se emplea el valor nulo (NULL).
 - **Nulo** no representa 0 o una cadena vacía, **representa** un **valor** en sí **mismo** (valor desconocido).
 - **No hay dos valores nulos iguales.**
 - Pueden dificultar el tratamiento de los datos y se recomienda usar con precaución.

Orígenes del Modelo Relacional

Restricciones del Modelo Relacional

- **Restricción de entidad (Primary Key):**
 - La **PK** debe contener **valores únicos y no nulos**

ALUMNO				
<u>DNI</u>	Nombre	Apellidos	Grado	Facultad
73453212N	Juan	Rodríguez Arias	Robótica	EPSE
39567348H	Ana	Blanco García	Paisaxe	EPSE
29547206J	María	PrieTo Estévez	Civil	EPSE
39567348H	Luis	Méndez Ríos	ADE	ADE
NULL	Maite	Gómez Pérez	EyT	ADE

¿Cómo se sabe cual es el alumno correcto si los dos tienen el mismo DNI?

Cómo se identifica a Maite si no tenemos el DNI?

Orígenes del Modelo Relacional

Restricciones del Modelo Relacional

● Restricción de entidad referencial:

- Los **valores** de la **FK** deben tener el **mismo dominio** que los **valores** de la **PK** que **referencian**

CLIENTE		
<u>NIF</u>	Nombre	Apellidos
73453212N	María	Rodríguez Arias
39567348H	Juan	Prieto Estévez

PEDIDOS		
<u>NIF</u>	<u>Cod. Prod</u>	Unidades
73453212N	AABBC	2
XS23FR11G	111222	1

PRODUCTO		
<u>Codigo</u>	Nombre	Precio
111111	Portátil	1500
111222	Impresora	120

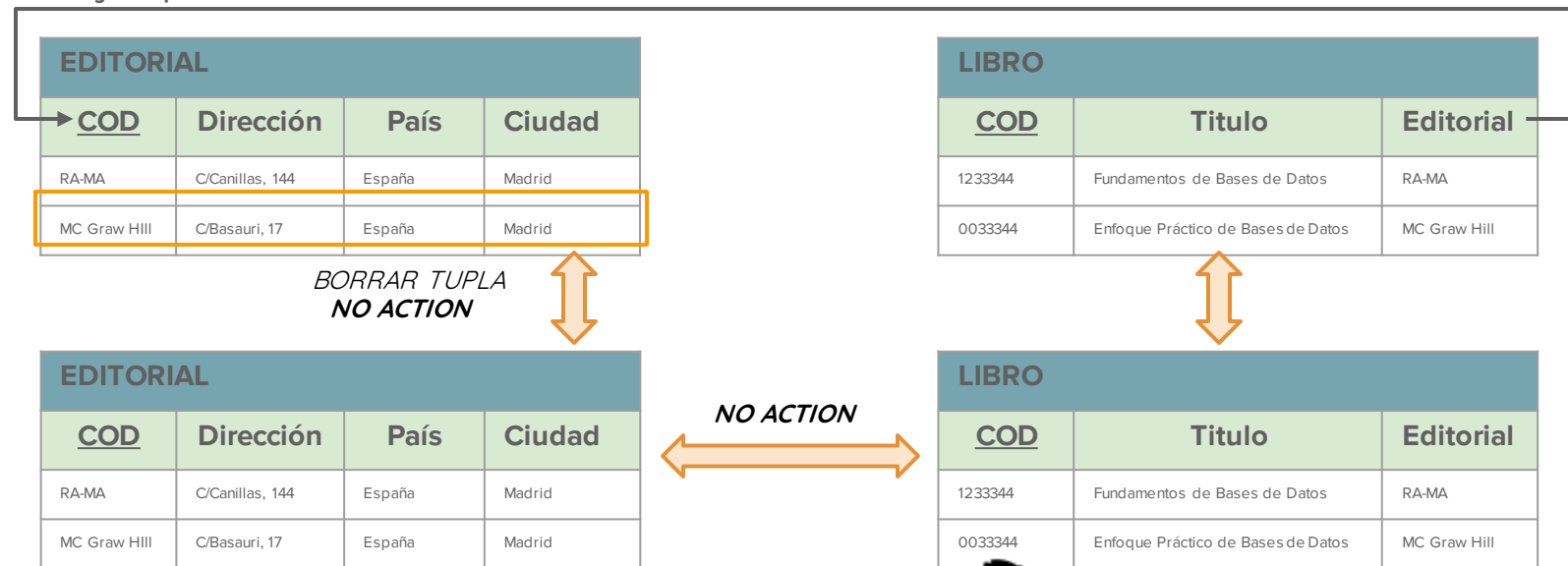
- Permiten determinar las consecuencias de las operaciones de borrado y modificación sobre la relación referenciada
 - **NO ACTION:** sólo se permiten modificar las PK que no estan referenciadas por FKs.
 - **CASCADE:** propaga la acción realizada en la PK a las FKs que referencia.
 - **SET NULL:** pone como valor NULL las FK alteradas tras una operación sobre las PKs.
 - **SET DEFAULT:** asigna un valor por defecto a las FK que referencia a las PKs que han sido modificadas/borradas.

Orígenes del Modelo Relacional

Restricciones del Modelo Relacional

- **Restricción de entidad referencial:**

- Ejemplo:



¿Porque no se modifican las relaciones?

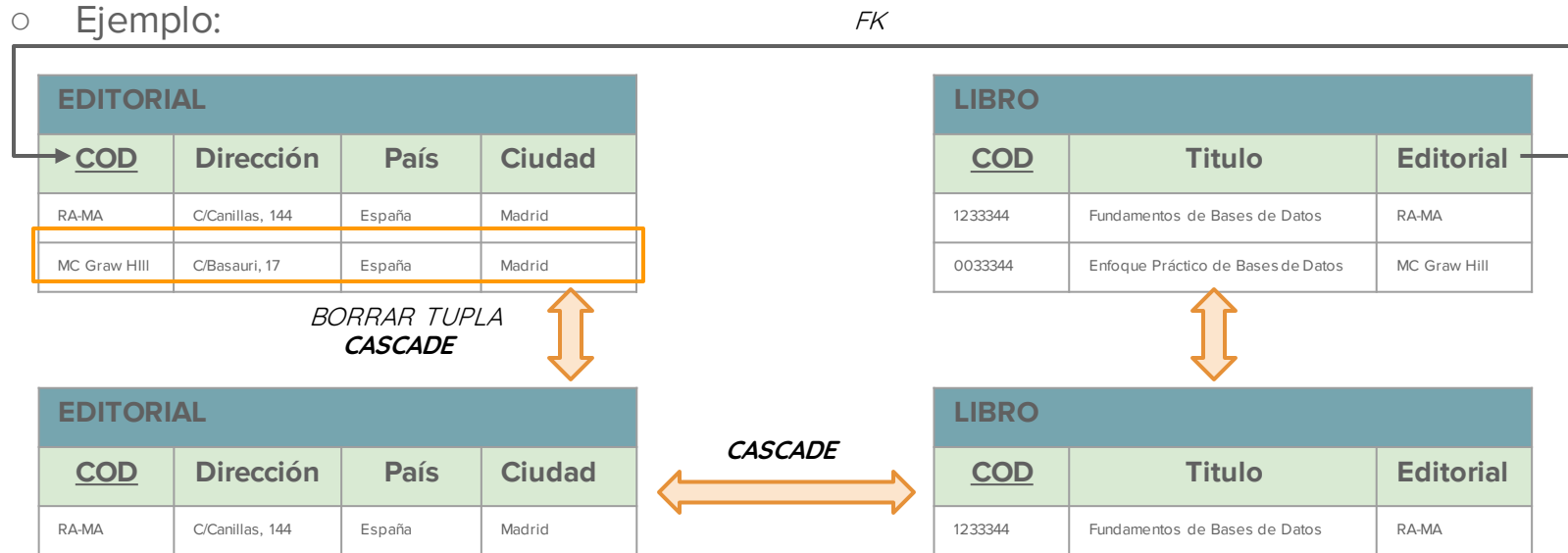


Orígenes del Modelo Relacional

Restricciones del Modelo Relacional

- **Restricción de entidad referencial:**

- Ejemplo:

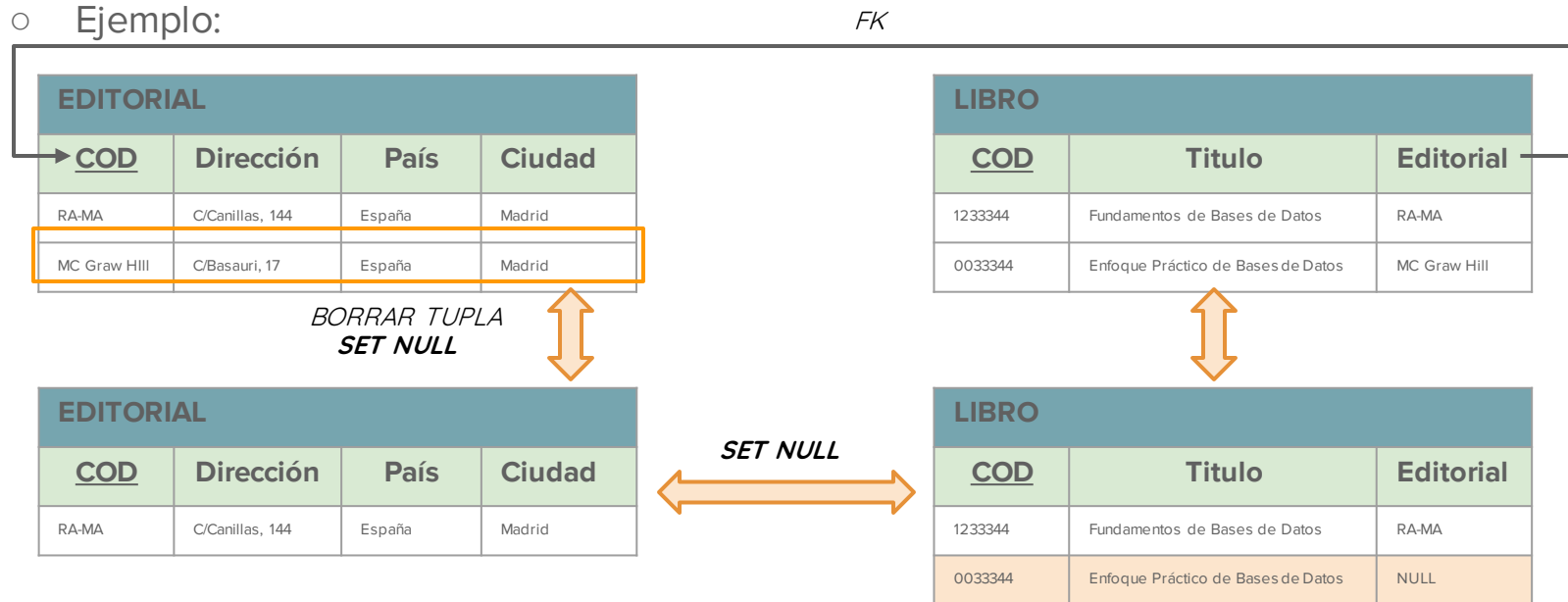


Orígenes del Modelo Relacional

Restricciones del Modelo Relacional

- **Restricción de entidad referencial:**

- Ejemplo:

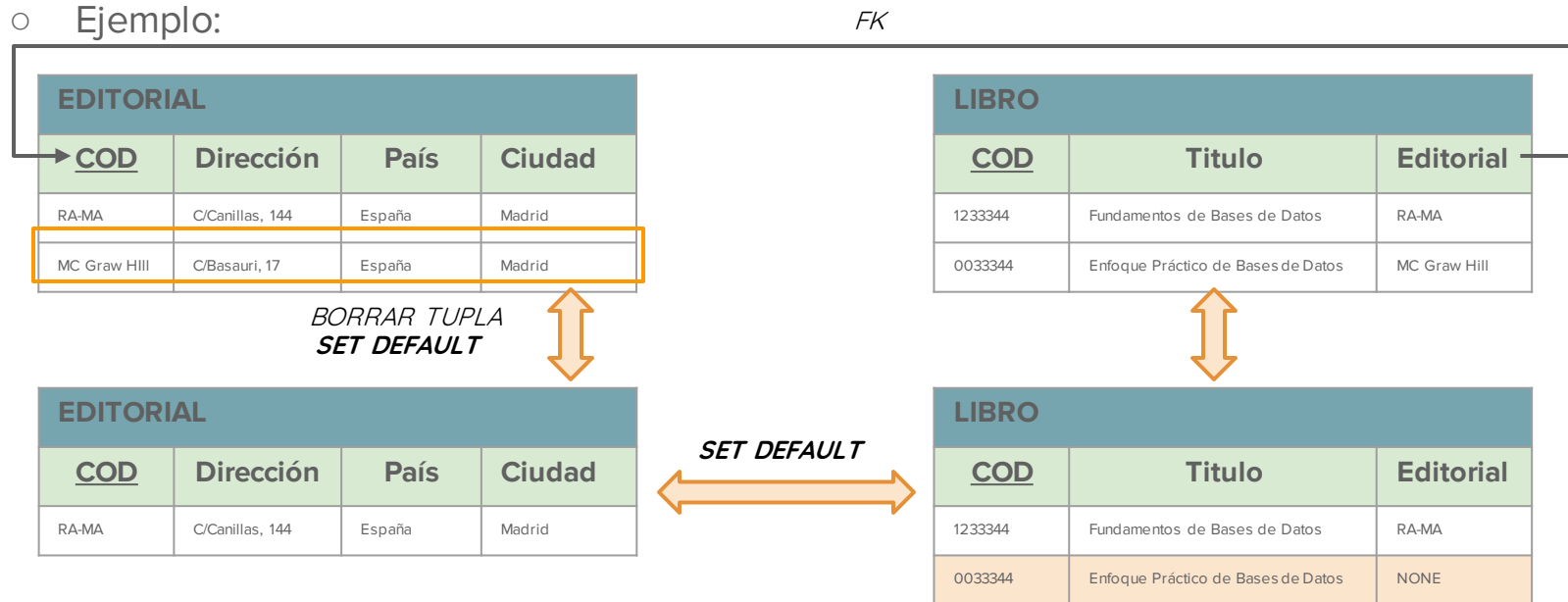


Orígenes del Modelo Relacional

Restricciones del Modelo Relacional

- **Restricción de entidad referencial:**

- Ejemplo:



Orígenes del Modelo Relacional

Restricciones del Modelo Relacional

● De usuario:

- **Condición** sobre un conjunto de atributos $\{A_1, A_2, \dots, A_n\}$, tuplas $\{t_1, t_2, \dots, t_n\}$ o dominios $\{D_1, D_2, \dots, D_n\}$ que debe ser **verificada** en toda operación de **actualización**.

- **Verificación (CHECK):** comprueba si el predicado (valor) es cierto (cumple la restricción) o falso (incumple la restricción) y aborta la operación.

- Sólo puede afectar a un elemento
- Puede tener nombre

```
CHECK FECHA_NAC < 1990
```

- **Aserción (ASSERTION):** Similar a la anterior, pero puede afectar a varios elementos.

- Siempre debe tener un nombre

```
CREATE ASSERTION solo_un_presi CHECK  
(SELECT COUNT(*) FROM empleado WHERE puesto='PRESIDENTE') < 2)
```

Orígenes del Modelo Relacional

Restricciones del Modelo Relacional

● Disparadores (Triggers):

- Restricción en la que el usuario puede especificar la respuesta (acción) ante una determinada condición.
- Existen dos tipos:
 - **BEFORE:** se ejecuta el disparador antes de realizar la condición.
 - **AFTER:** se ejecuta el disparador después de realizar la condición.

```
CREATE TRIGGER increment_animal
  AFTER INSERT ON animals FOR EACH ROW
  UPDATE animal_count SET animal_count.animals = animal_count.animals+1;
```

¿Que hace?



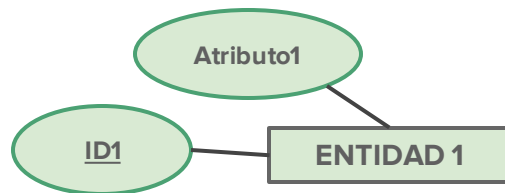
● Dependencia Funcional (DF):

- Establece una relación de dependencia entre dos conjuntos de atributos X e Y

Conversión del Modelo ER a Modelo Relacional

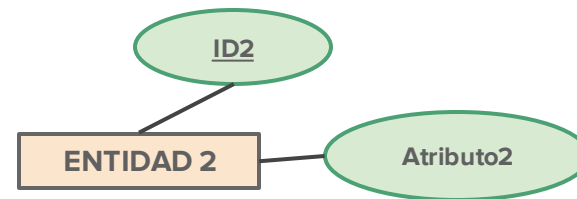
Entidades

- Cada entidad del modelo E-R genera una tabla. Dicha tabla contiene como columnas cada uno de los atributos de la entidad. Además puede contener otras columnas fruto de relaciones, normalmente 1:N, con otras entidades.



ENTIDAD 1

<u>ID1</u>	Atributo1



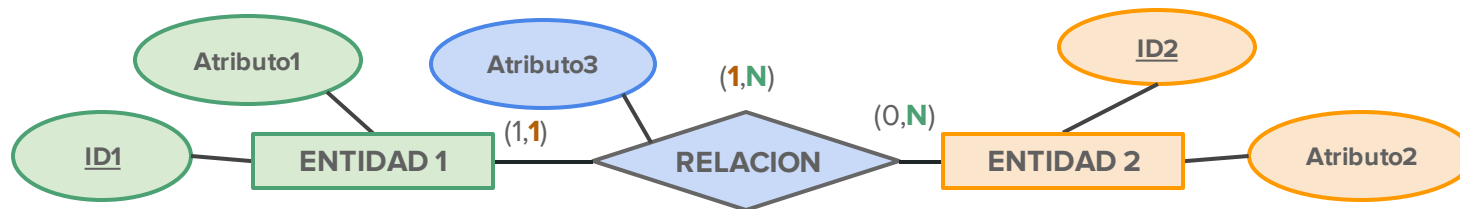
ENTIDAD 2

<u>ID2</u>	Atributo2

Conversión del Modelo ER a Modelo Relacional

Relacion 1:N (entidades fuertes)

- Se genera una tabla por cada entidad y se traslada la clave principal la tabla del lado 1 al lado N en el que se convierte en clave foránea. Si la relación tiene atributos se envían a la tabla del lado N.



ENTIDAD 1

<u>ID1</u>	Atributo1

ENTIDAD 2

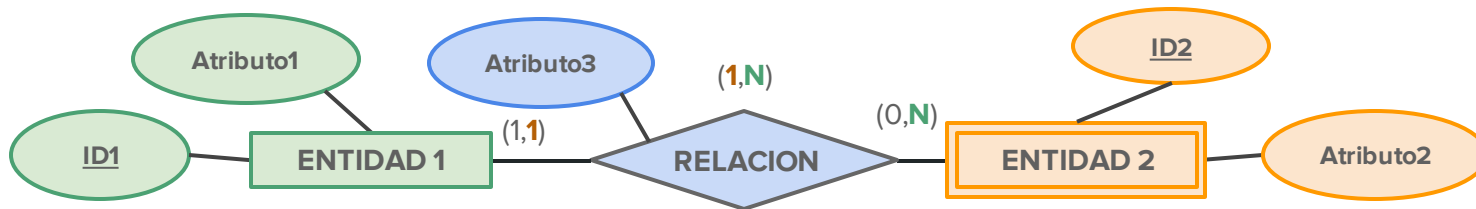
<u>ID2</u>	Atributo2	<u>ID1</u>	Atributo3

FK

Conversión del Modelo ER a Modelo Relacional

Relacion 1:N (entidades débiles)

- Caso 1 - entidad débil con clave primaria: se genera una tabla por cada entidad y se traslada la clave principal la tabla del lado 1 al lado N en el que se convierte en clave foránea. Si la relación tiene atributos se envían a la tabla del lado N.



ENTIDAD 1

<u>ID1</u>	Atributo1

ENTIDAD 2

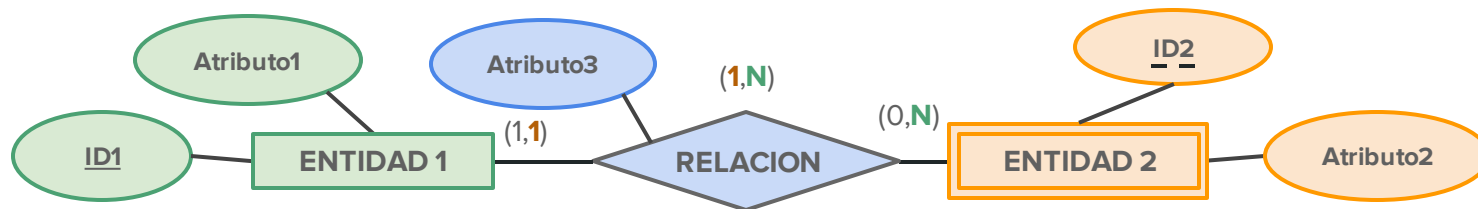
<u>ID2</u>	Atributo2	<u>ID1</u>	Atributo3

FK

Conversión del Modelo ER a Modelo Relacional

Relacion 1:N (entidades débiles)

- Caso 2 - entidad débil sin clave primaria: se genera una tabla por cada entidad y se traslada la clave principal la tabla del lado 1 al lado N, formando clave primaria compuesta con el atributo discriminante. Si la relación tiene atributos se envían a la tabla del lado N.



ENTIDAD 1

<u>ID1</u>	Atributo1

ENTIDAD 2

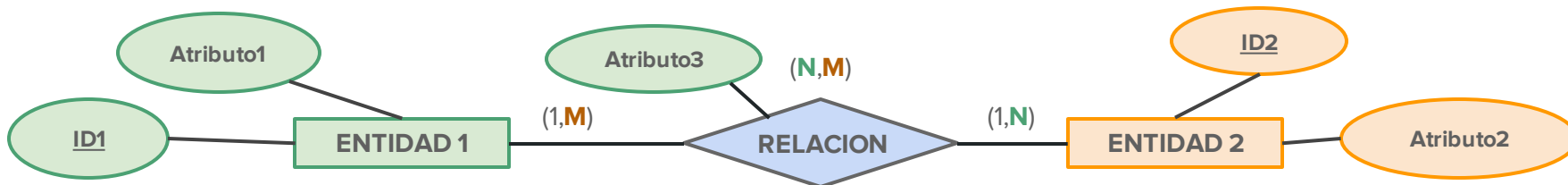
<u>ID1</u>	ID2	Atributo2	Atributo3

FK

Conversión del Modelo ER a Modelo Relacional

Relación N:M

- Se crea una nueva relación con los atributos propios (si los hubiese). La clave primaria estará compuesta por las PKs de las tablas participantes en la relación.



ENTIDAD 1

<u>ID1</u>	Atributo1

RELACION

<u>ID1</u>	<u>ID2</u>	Atributo3

ENTIDAD 2

<u>ID2</u>	Atributo2

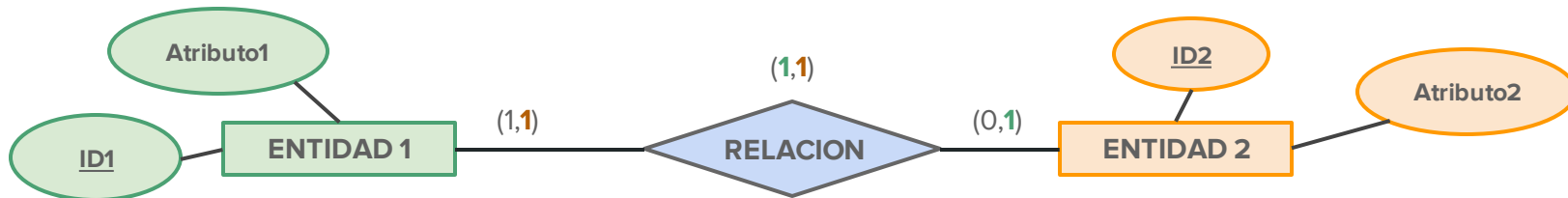
FK

FK

Conversión del Modelo ER a Modelo Relacional

Relación 1:1 (entidades fuertes)

- Se genera una tabla por cada entidad y se coloca la clave principal de una tabla en la otra tabla sólo como clave foránea (FK). Se recomienda llevar la PK a la tabla de la relación 0 (si la hay)



ENTIDAD 1

<u>ID1</u>	Atributo1

ENTIDAD 2

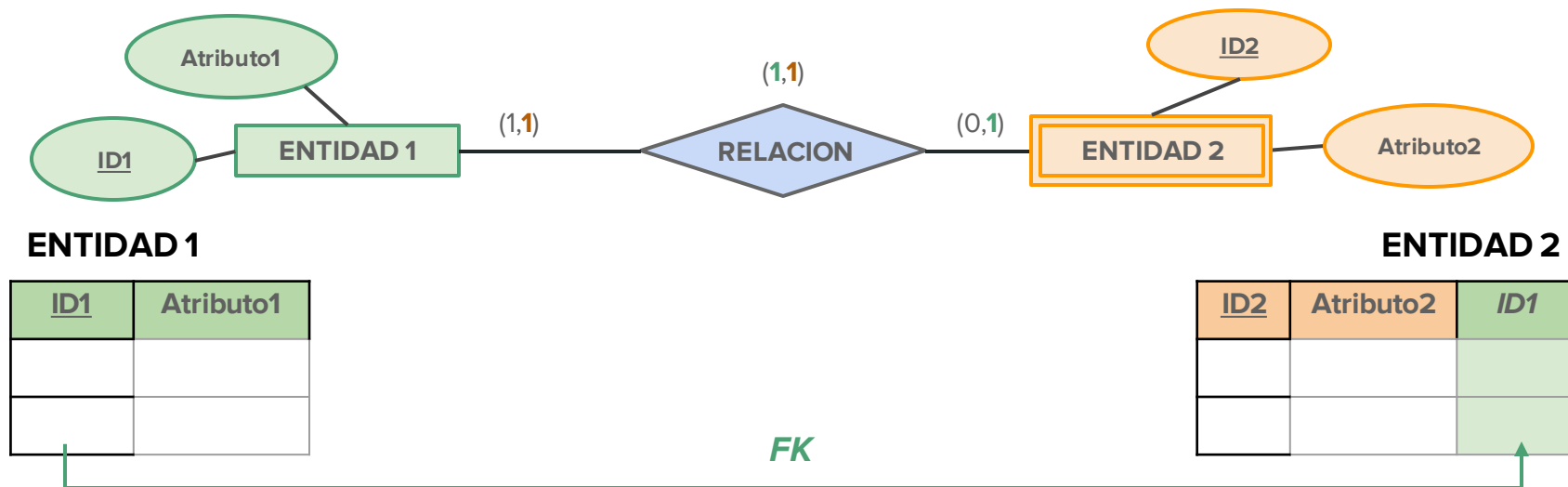
<u>ID2</u>	Atributo2	<u>ID1</u>

FK

Conversión del Modelo ER a Modelo Relacional

Relación 1:1 (entidad débil)

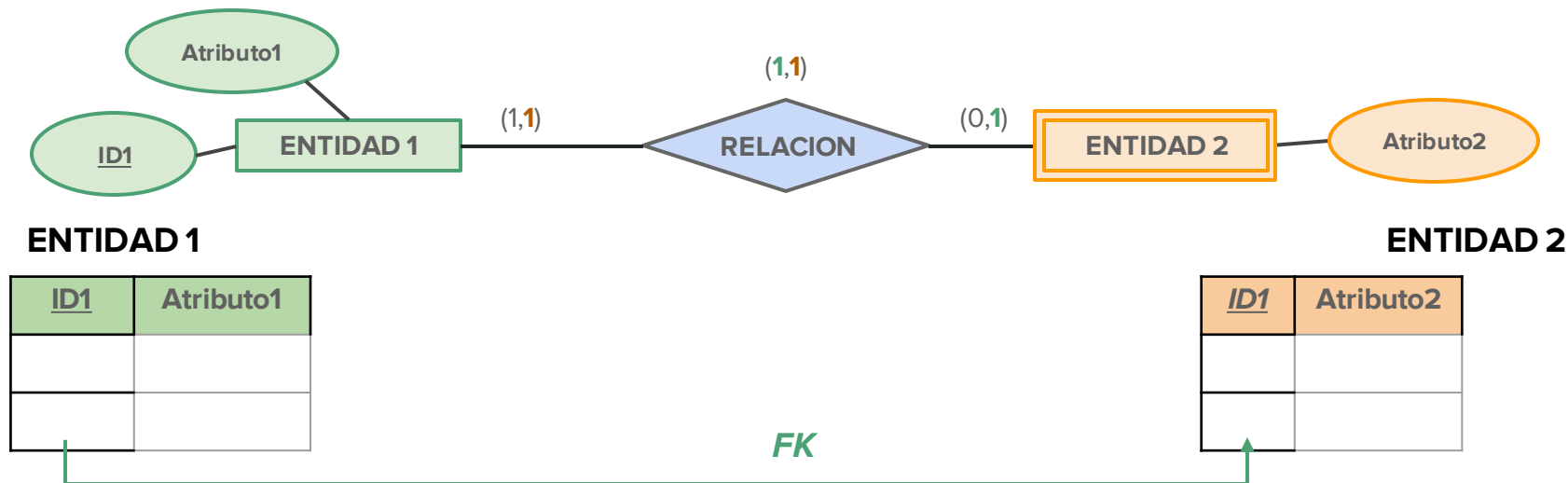
- Caso 1 - entidad débil con clave primaria: se genera una tabla por cada entidad y se coloca la clave principal de la entidad fuerte en la otra tabla sólo como clave foránea (FK).



Conversión del Modelo ER a Modelo Relacional

Relación 1:1 (entidad débil)

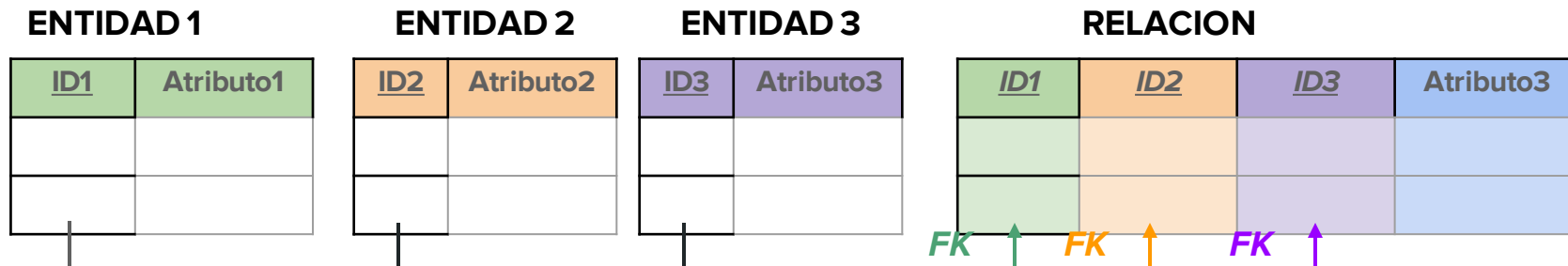
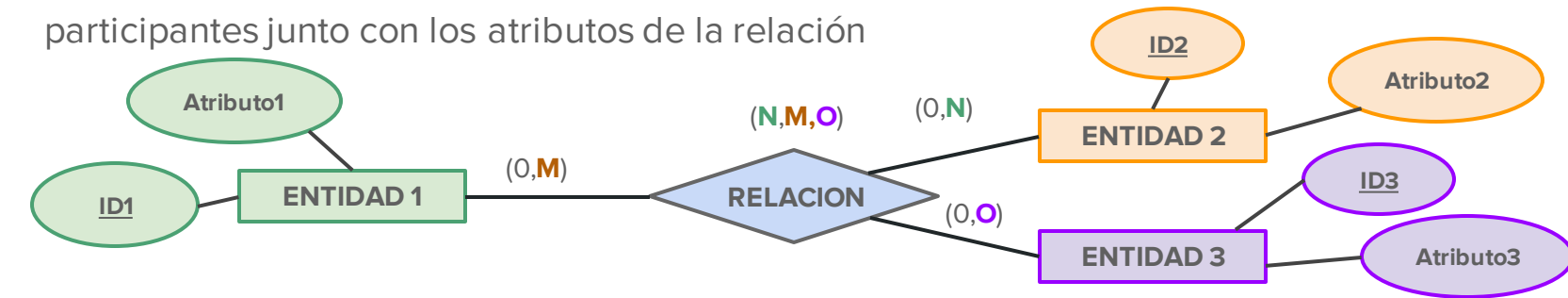
- Caso 2 - entidad débil sin clave primaria: se genera una tabla por cada entidad y se coloca la clave principal de la entidad fuerte en la de la entidad débil como clave primaria y foránea.



Conversión del Modelo ER a Modelo Relacional

Relación N:M:O

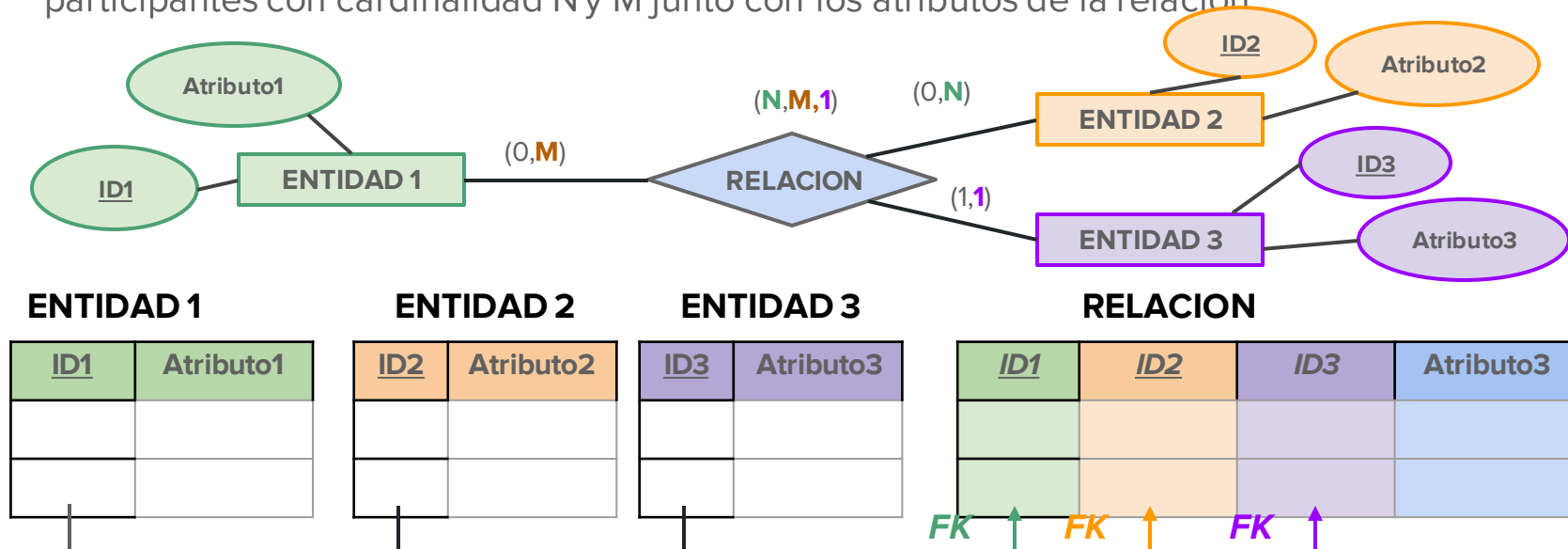
- Se genera una tabla por cada entidad y otra para la relación ternaria. La tabla de la relación ternaria tendría una clave primaria compuesta por las claves primarias de las entidades participantes junto con los atributos de la relación



Conversión del Modelo ER a Modelo Relacional

Relación 1:N:M

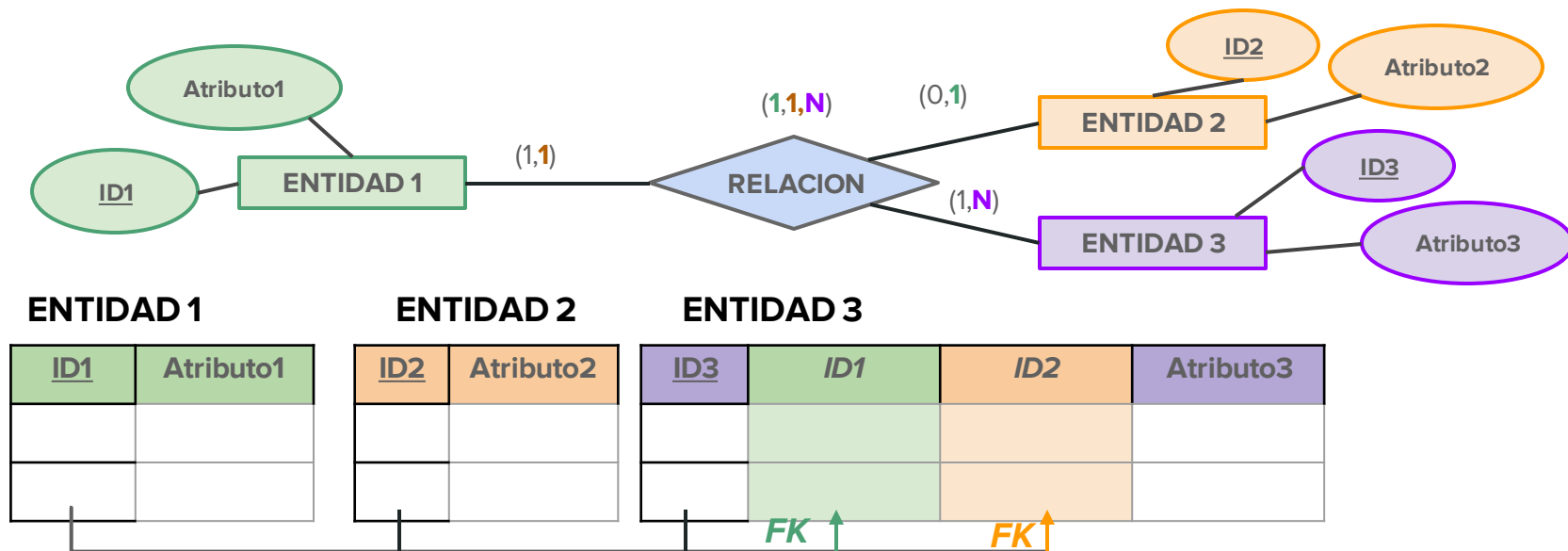
- Se genera una tabla por cada entidad y otra para la relación ternaria. La tabla de la relación ternaria tendría una clave primaria compuesta por las claves primarias de las entidades participantes con cardinalidad N y M junto con los atributos de la relación



Conversión del Modelo ER a Modelo Relacional

Relación 1:1:N

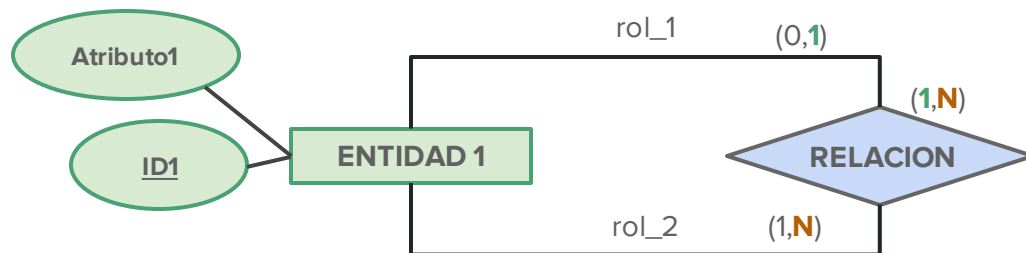
- Se genera una tabla por cada entidad. La tabla de la relación N tendría como claves foráneas las PKs de las otras tablas pero no sería PK en esa tabla.



Conversión del Modelo ER a Modelo Relacional

Relación recursiva

- Caso 1 - Relación 1:N: Crear una tabla con todos los atributos replicando la PK con un nombre nuevo que es clave foránea de la propia tabla.



Cuidado: la FK debe referenciar a la PK de la propia tabla!!



ENTIDAD 1

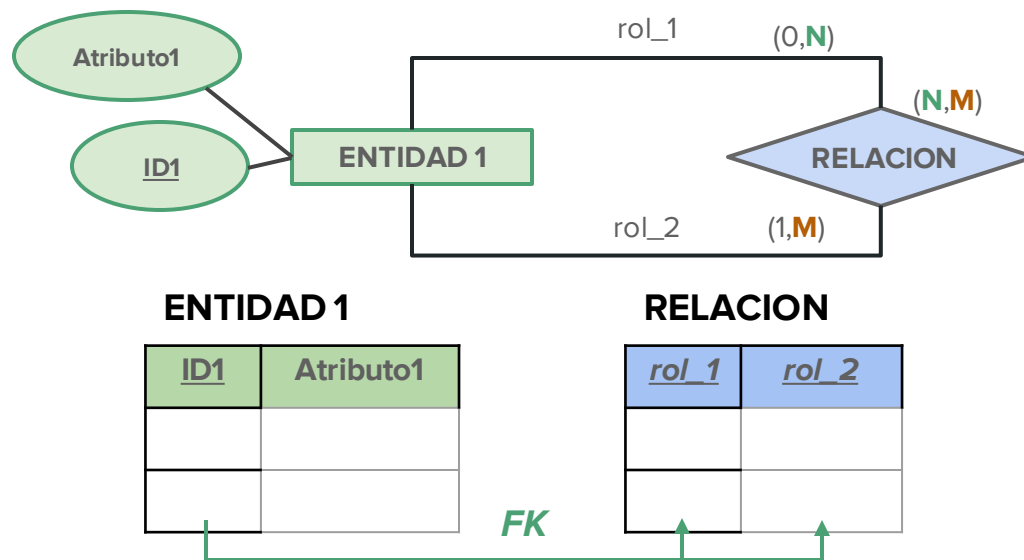
<u>ID1</u>	Atributo1	rol_1

FK

Conversión del Modelo ER a Modelo Relacional

Relación recursiva

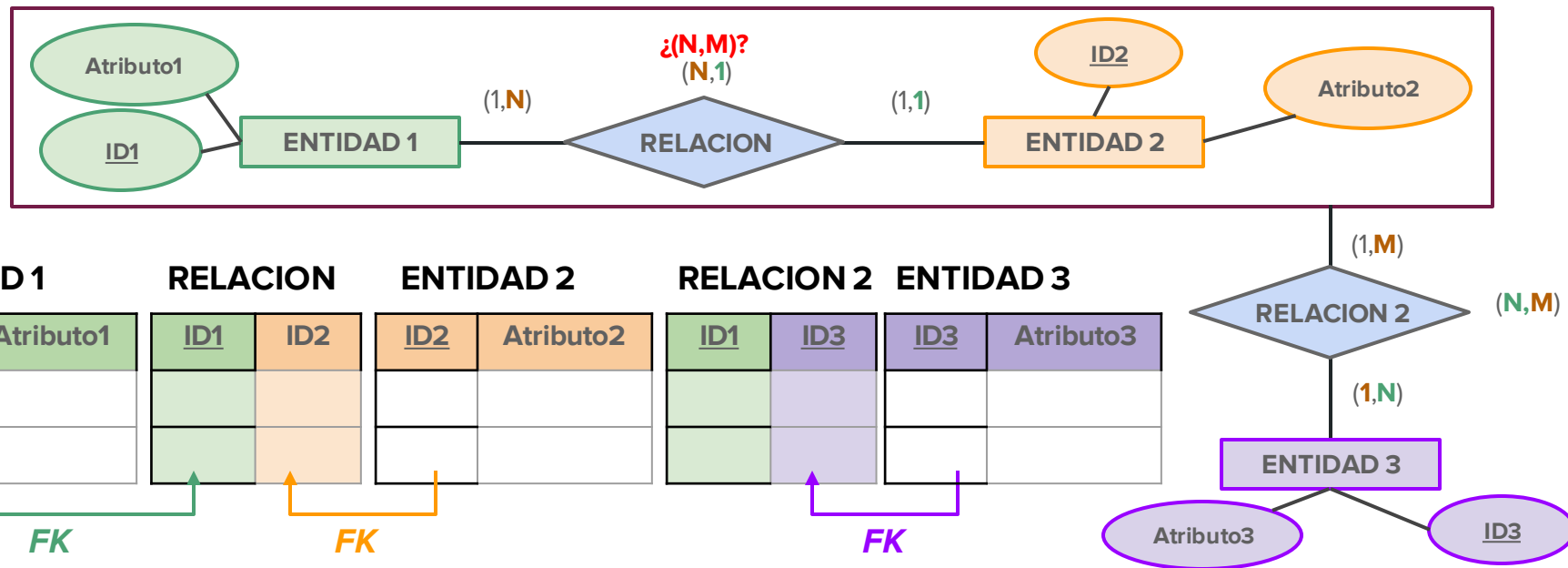
- Caso 2 - Relación N:M: Crear una tabla para la entidad y otro para la relación. La tabla de la relación tendrá como clave compuesta los atributos de los roles que a su vez serán clave foránea de la entidad.



Conversión del Modelo ER a Modelo Relacional

Relación Agregación

- Se trata la relación dentro de la categorización como si fuera una entidad (E). Las entidades que se relacionan con la categorización estarán relacionadas con E .



Conversión del Modelo ER a Modelo Relacional

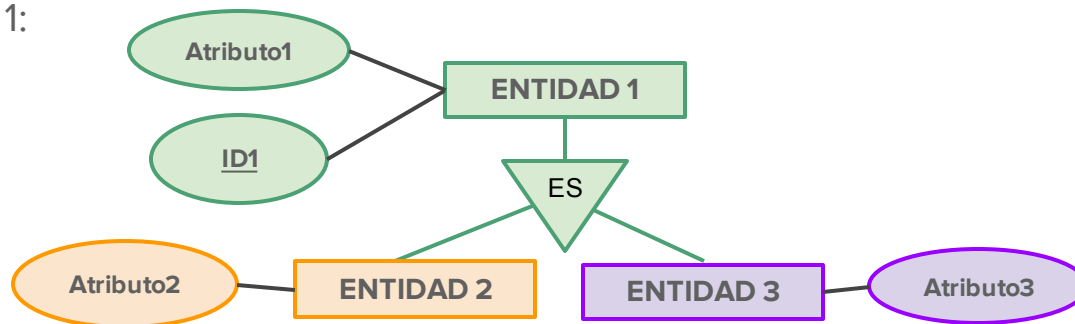
Relación Generalización/Especialización

- Método 1: crear un esquema por cada entidad
 - Recomendado cuando:
 - Existen muchos atributos distintos entre los subtipos
 - Se quiere mantener de todas maneras los atributos comunes a todos en una relación
 - Es **solapada parcial**
 - Inconvenientes:
 - La obtención de información requiere el acceso a dos relaciones
 - Bajo nivel (subclases)
 - Alto nivel (superclase)
 - La opción más común.

Conversión del Modelo ER a Modelo Relacional

Relación Generalización/Especialización

- Método 1:



ENTIDAD 1

<u>ID1</u>	Atributo1

ENTIDAD 2

<u>ID1</u>	Atributo2

ENTIDAD 3

<u>ID1</u>	Atributo3



Conversión del Modelo ER a Modelo Relacional

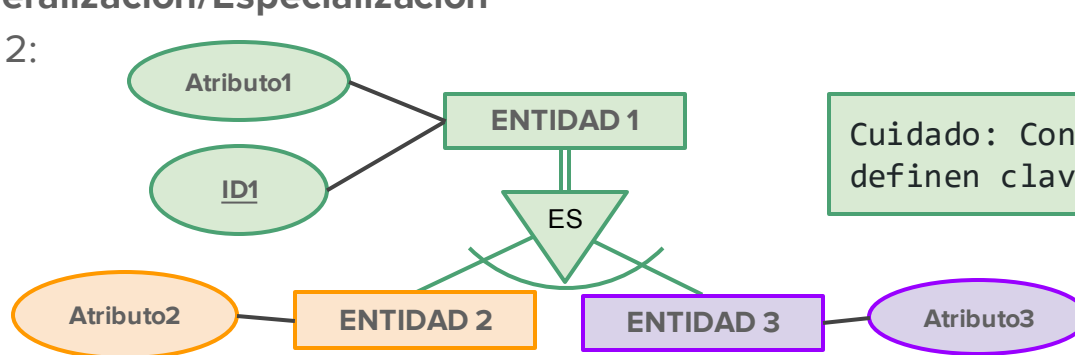
Relación Generalización/Especialización

- Método 2: crear un esquema por cada conjunto de entidades hijas.
 - Recomendado cuando:
 - Es **total disjunta**: la entidad padre no requiere que se almacene información.
 - Muchos atributos distintos en las entidades hijas.
 - Inconvenientes:
 - Si es total solapada se repetirían atributos (redundancia).
 - Ejemplo: calle y ciudad se repetirían para personas que sean clientes y empleados.

Conversión del Modelo ER a Modelo Relacional

Relación Generalización/Especialización

- Método 2:



Cuidado: Con este método no se definen claves foráneas



ENTIDAD 2

<u>ID1</u>	Atributo1	Atributo2

ENTIDAD 3

<u>ID1</u>	Atributo1	Atributo3

Conversión del Modelo ER a Modelo Relacional

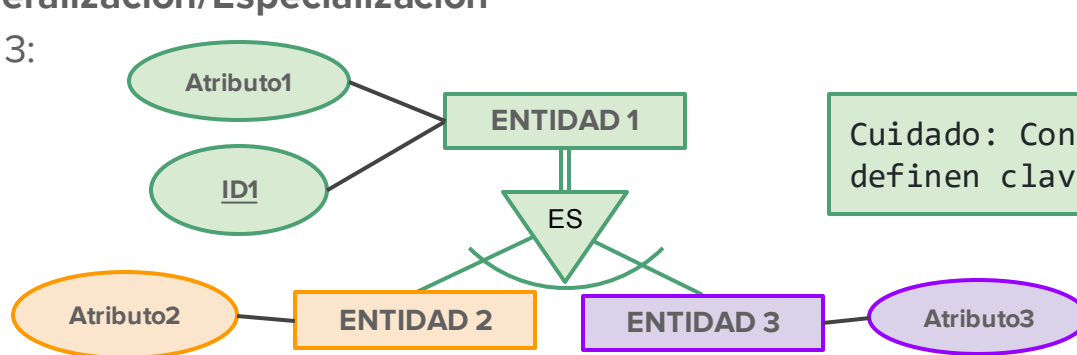
Relación Generalización/Especialización

- Método 3: crear un esquema que contenga los atributos de la clase padre y todas las clases hijas.
 - Recomendado cuando:
 - Los subtipos se diferencian en muy pocos atributos (o ninguno).
 - Compatible con relaciones de participación total o parcial.
 - Inconvenientes:
 - No permite relaciones solapadas ya que se tendrían tuplas con la misma PK.

Conversión del Modelo ER a Modelo Relacional

Relación Generalización/Especialización

- Método 3:

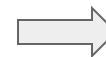


Cuidado: Con este método no se definen claves foráneas



ENTIDAD 1

<u>ID1</u>	Atributo1	Atributo2	Atributo3	tipo



Si hay un atributo en la relación, se pone el de la relación. Sino, se añade uno nuevo

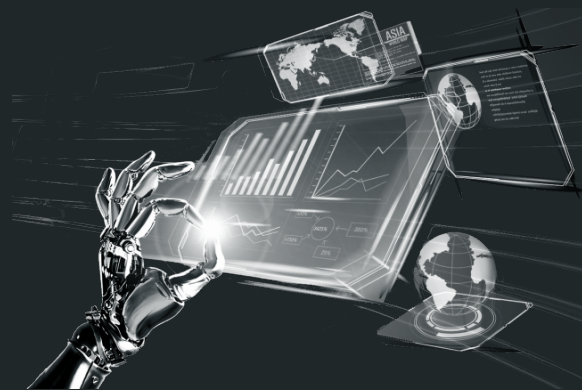


Bibliografía

Ramez A. Elmasri, Shamkant B. Navathe. Fundamentos de Sistemas de Bases de Datos (5º edic.).

Prentice-Hall. 2007, [Cap. 6]

de Miguel, A.; Piattini, M. Fundamentos y modelos de bases de datos (2ª edic. Ra-Ma, 1999, [Cap. 5]



Gestión de Datos para Robótica

T1d - Modelo Relacional

Álvaro Vázquez Álvarez
Departamento de Electrónica e Computación

✉ alvaro.vazquez@usc.es

📍 Pabellón III - Despacho 4

Curso 2023-2024