

MODELO LOGICO RELACIONAL

Estructura del Modelo Relacional

Las bases de datos relacionales se basan en un modelo lógico conocido como modelo relacional. Trabajamos **únicamente** con relaciones

Para relacionar != tablas, tenemos que usar **Datos** que relacionan las tablas, no índices de fila o cosas así. **Para hacer las referencias usamos datos.**

Otra cosa importante es evitar que los datos tengan múltiples valores. No definir datos que puedan tener una estructura compleja de los valores, sino que sean **atómicos**. (si tengo una fecha y quiero comparar por fecha no pasa nada, y si quiero solo comparar por día ahí ya no sería atómico y me puedo guardar solo el día)

Algunas definiciones:

- **Dominio:** Es un conjunto de valores homogéneos.
Ejemplo: $D1 = \{\text{Barcelona, Sevilla, Buenos Aires}\}$ $D2 = \{\text{Argentina, España, Chile}\}$
- **Producto cartesiano:** $A \times B$ se define como el conjunto de pares (a,b) que cumplen que $a \in A$ y $b \in B$.
En nuestro ejemplo $D1 \times D2 = \{(\text{Barcelona, Argentina}), (\text{Barcelona, España}), (\text{Barcelona, Chile}), (\text{Sevilla, Argentina}), (\text{Sevilla, España}), (\text{Sevilla, Chile}), (\text{Buenos Aires, Argentina}), (\text{Buenos Aires, España}), (\text{Buenos Aires, Chile})\}$
- **Relación:** es un subconjunto de un producto cartesiano. En el ejemplo anterior podría especialmente interesarnos la siguiente relación: $R = \{(\text{Barcelona, España}), (\text{Sevilla, España}), (\text{Buenos Aires, Argentina})\}$
Acá me podría estar dando la relación: ciudad que pertenece a un país.

Formalización

En el modelo relacional, un nombre de relación R junto con una lista de atributos asociados se denomina **esquema de relación**, y se denota de la siguiente manera: $R(A1,A2,...,An)$

Ejemplo: Películas(nombre_película, año, nombre_director, cant_oscars)

Cada uno de los atributos A_i está asociado a un dominio particular, $\text{dom}(A_i)$. En nuestro ejemplo: $\text{nombre_película} \rightarrow \text{dom}(\text{nombre_película}) = \text{string}$ $\text{año} \rightarrow \text{dom}(\text{año}) = \mathbb{N}^+$, etc.

Relaciones y tuplas

Una relación R con esquema de relación $R(A1,A2,...,An)$, estando los atributos A_i asociados a los dominios $D_i = \text{dom}(A_i)$ es un subconjunto del producto cartesiano $D1 \times D2 \times \dots \times Dn$.

Dado el esquema de relación *Películas(nombre_película, año, nombre_director, cant_oscars)*, se cumple que:

$\text{Películas} \subset \text{dom}(\text{nombre_película}) \times \text{dom}(\text{año}) \times \text{dom}(\text{nombre_director}) \times \text{dom}(\text{cant_oscars})$

Ejemplo: Películas = $\{(\text{Kill Bill, 2003, Quentin Tarantino, 0}), (\text{Django Unchained, 2012, Quentin Tarantino, 2}), (\text{Star Wars III, 2005, George Lucas, 0}), (\text{El Cisne Negro, 2010, Darren Aronofsky, 1})\}$

Un elemento de una relación se denomina tupla. $t = (\text{Kill Bill, 2003, Quentin Tarantino, 0})$ es una tupla de Películas. Lo denotamos como $t \in \text{Películas}$. $u = (\text{Kill Bill, 2003, Quentin Tarantino, 3})$ no es una tupla de Películas porque $u \notin \text{Películas}$ (yo conozco el Dominio y se que el 3 no va, es falso)

Ejemplo

Si pensamos a una relación como un predicado, podemos predicar que Películas(Kill Bill,2003,Quentin Tarantino,3) = **F**.

El **valor** tomado por un atributo A en la tupla t se denota $t[A]$ o $t.A$.

- $t[\text{nombre_director}] = t.\text{nombre_director} = \text{Quentin Tarantino}$
- $t[\text{nombre_pelicula}, \text{año}] = t.(\text{nombre_pelicula}, \text{año}) = (\text{Kill Bill}, 2003)$.

La **cardinalidad** de una relación R es la cantidad de tuplas que posee. La simbolizaremos $n(R)$. Ejemplo: $n(\text{Películas}) = 4$.

Tablas

Una forma útil de representar una relación es a través de una tabla:

- *Columnas*: los atributos de la relación
- *Filas*: representan las tuplas. En el caso de la relación Películas:

nombre_película	año	nombre_director	cant_oscars
Kill Bill	2003	Quentin Tarantino	0
Django Unchained	2012	Quentin Tarantino	2
Star Wars III	2005	George Lucas	0
El Cisne Negro	2010	Darren Aronofsky	1

Los siguientes conceptos son equivalentes:

- Archivos -> tablas
- Registros -> filas
- Campos -> columnas

Restricciones

Las relaciones del modelo relacional representan generalmente entidades o interrelaciones de nuestro modelo de datos. Deben cumplir una serie de restricciones de distintos tipos. Me sirven para indicar cómo son los datos que estamos modelando. Yo quiero que en mis tablas no halla datos inválidos, y si yo quiero ingresar un dato que no cumple, no me deje ingresarlo.

Restricciones de dominio

Dado un atributo A de una relación R, el valor del atributo en una tupla t debe pertenecer al dominio $\text{dom}(A)$.

Algunos dominios posibles: N, N+, R, R+. caracter, string, valor booleano, fecha, conjunto finito (categorías).

- Se puede permitir que algunos de los atributos tomen un valor nulo (**NULL**).
- Los atributos deben ser atómicos (no se permiten atributos compuestos o multivaluados).
- No pueden existir dos tuplas distintas que coincidan en los valores de todos sus atributos.

Superclave: existe un subconjunto SK del conjunto de atributos (A_1, A_2, \dots, A_n) de R tal que dadas dos tuplas $s, t \in R$, las mismas difieren en al menos uno de los atributos de SK

Ejemplo: ISBN es superclave de libros. {ISBN, Nombre, año} también es superclave

- Nos interesan las superclaves **mínimas**: no admiten ningún subconjunto propio con la misma propiedad. A estas superclaves las llamaremos **claves candidatas** o simplemente claves. ISBN es clave candidata.
- De entre todas las claves candidatas elegiremos una como **clave primaria** de la relación. La indicaremos subrayada en el esquema.

Ejemplo

Películas(nombre_película, año, nombre_director, cant_oscars)

- Si suponemos que no puede haber dos películas con el mismo nombre
 - “nombre_película” es clave candidata.
 - Es la única. La designaremos como clave primaria.
 - Lo representamos como: Películas(nombre_película, año, nombre_director, cant_oscars).
 - {nombre_película, cant_oscars} es superclave, pero no es clave candidata porque no es minimal.
- Si admitimos que existen películas distintas con el mismo nombre
 - Deberíamos crear un atributo “id” que nos permita identificar a cada película.
 - El esquema de relación sería ahora:
Películas(id, nombre_película, año, nombre_director, cant_oscars)
 - “id” será la clave primaria.

Agregarle un *id* a una tabla se conoce como **subrogated key**

Veamos otro ejemplo

ActoresPelículas(nombre_película, año, nombre_director, cant_oscars, nombre_actor)

nombre_película	año	nombre_director	cant_oscars	nombre_actor
Kill Bill	2003	Quentin Tarantino	0	Uma Thurman
Django Unchained	2012	Quentin Tarantino	2	Leonardo DiCaprio
Django Unchained	2012	Quentin Tarantino	2	Samuel L. Jackson
Star Wars III	2005	George Lucas	0	Samuel L. Jackson
Star Wars III	2005	George Lucas	0	Natalie Portman
El Cisne Negro	2010	Darren Aronofsky	1	Natalie Portman

- Ya no alcanza “nombre_película” para identificar una tupla.
- La clave de la relación ActoresPelículas es {nombre_película, nombre_actor}!

Nota: Desde ya, este es un diseño desprolijo, por contener información redundante sobre las películas en distintas tuplas.

Agregamos nombre_de_actor. Como el atributo tiene que ser atómico (solo 1 valor) si tengo una peli con muchos actores debería tener varias filas. Ahora, eso en la misma tabla queda horrible. Es un diseño desprolijo con información redundante.

Para solucionar esto, utilizaremos esquemas de bases de datos.

Esquemas de bases de datos

Las bases de datos almacenan múltiples esquemas de relación, muchas veces relacionados entre ellos.

En el modelo relacional, una base de datos se representa a través de un esquema de base de datos relacional.

Esquema de base de datos relacional S: es un conjunto de esquemas de relación $S = \{R1, R2, \dots, Rm\}$ junto con una serie de restricciones de integridad.

Ejemplo: Cine = {Películas, Actores, Actuaciones}, en donde:

- Películas (nombre_película, año, nombre_director, cant_oscars)
- Actores (nombre_actor, país)
- Actuaciones (nombre_película, nombre_actor)

Actuaciones es lo que vincula a las Películas con los Actores.

Ejemplo de instancia de nuestra BDD Cine:

PELÍCULAS			
nombre_pelicula	año	nombre_director	cant_oscars
Kill Bill	2003	Quentin Tarantino	0
Django Unchained	2012	Quentin Tarantino	2
Star Wars III	2005	George Lucas	0
El Cisne Negro	2010	Darren Aronofsky	1

ACTORES		ACTUACIONES	
nombre_actor	país	nombre_pelicula	nombre_actor
Uma Thurman	Estados Unidos	Kill Bill	Uma Thurman
Leonardo DiCaprio	Estados Unidos	Django Unchained	Leonardo DiCaprio
Samuel L. Jackson	Estados Unidos	Django Unchained	Samuel L. Jackson
Natalie Portman	Israel	Star Wars III	Samuel L. Jackson
		Star Wars III	Natalie Portman
		El Cisne Negro	Natalie Portman

Lo que si, mi motor NO me puede dejar agregar cualquier cosa.

- En películas no puedo agregar una peli que ya este su nombre
- En Actores no puedo agregar un actor con un nombre que ya está
- En Actuaciones, no puedo volver a agregar el par Cisne Negro, Natalie Portman, y tampoco puedo poner la película Shrek, porque no está en la tabla Películas.

Restricciones de Integridad

Restricción de integridad de entidad: La clave primaria de una relación no puede tomar el valor nulo.

Restricción de integridad referencial: Cuando un conjunto de atributos FK de una relación R hace referencia a la clave primaria de otra relación S, entonces para toda tupla de R debe existir una tupla de S cuya clave primaria sea igual al valor de FK, a menos que todos los atributos de FK sean nulos.

Osea, si hay una referencia a una fila de otra tabla, el valor que usamos como referencia, tiene que estar en la otra tabla (en la columna referenciada)

Cine = {Películas, Actores, Actuaciones}

- Películas(nombre_pelicula, año, nombre_director, cant_oscars)
- Actores(nombre_actor, país)
- Actuaciones(nombre_pelicula, nombre_actor)

Ejemplo: Si una tupla en Actuaciones hace referencia “Star Wars III”, entonces debe existir “Star Wars III” en la relación Películas.

Sean $R(A1, A2, \dots, Ar)$ y $S(B1, B2, \dots, Bs)$ dos esquemas de relación. $FK \subset (A1, A2, \dots, Ar)$ hace referencia a S, cuya clave primaria es $PK \subset (B1, B2, \dots, Bs)$. Entonces:

$$\forall t \in R : t[FK] \neq NULL \rightarrow \exists s \in S : s[PK] = t[FK].$$

Lo que sí puede pasar es que la **fk sea completamente nula**, que pasa cuando no estoy referenciando nada de la otra tabla.

FK = clave foránea (Foreign Key) de S en R. Estas restricciones generalmente surgen de las interrelaciones entre entidades de nuestro modelo conceptual.

Cine = {Películas, Actores, Actuaciones}

Películas(nombre_pelicula, año, nombre_director, cant_oscars)

Actores(nombre_actor, país)

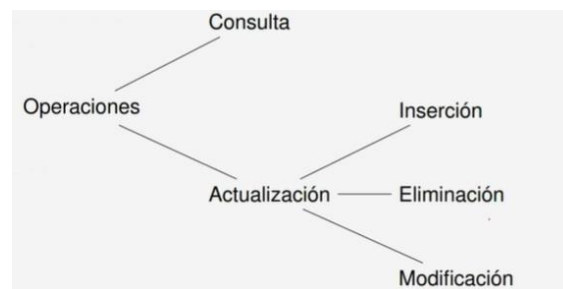
Actuaciones(nombre_pelicula, nombre_actor)

En nuestro ejemplo: Actuaciones.nombre_película es clave foránea y hace referencia a la relación Películas. Asimismo, Actuaciones.nombre_actor es clave foránea y hace referencia a la relación Actores. Indicaremos a las claves foráneas con un subrayado punteado.

Si las FKs vienen de tablas distintas, cuando las indique en el parcial las pongo por separado. Tipo: FKs = {nombre_pelicula}, {nombre_actor}.

Operaciones

Las operaciones del modelo relacional se especifican a través de lenguajes como el álgebra relacional o el cálculo relacional. Hay distintos tipos de operaciones



Operaciones de consulta

- No modifican ninguna relación existente.
- Por lo tanto no violan ningún tipo de restricción.

Operaciones de inserción de tuplas

- Pueden violar restricciones de dominio, de unicidad y de integridad de entidad o referencial.
- El SGBD debería rechazar una inserción que viola algún tipo de restricción.

Operaciones de eliminación

- Sólo pueden violar restricciones de integridad referencial (ej: si borro un actor en Actores y queda en Actuaciones).
- Cuando R referencia a S, y se intenta eliminar una tupla de S que es referenciada por alguna/s tupla/s en R.
- Hay tres estrategias posibles:
 1. rechazar la eliminación
 2. eliminar en cascada
 3. poner en NULL los atributos referenciales de las tuplas de R.

Operaciones de modificación

- Si se modifica una clave foránea, se debe verificar que sus nuevos valores referencien a una tupla existente de la relación referenciada, o bien sean todos nulos. De lo contrario se debería rechazar la operación.
- Si se modifica una clave primaria, puede violarse cualquiera de las restricciones de integridad, y se combinan las situaciones indicadas para inserción y eliminación.
- Si se modifica un atributo que no es ni FK ni PK, la única restricción que puede violarse es la de dominio

A veces es necesario realizar una serie de operaciones por completo, o bien no realizarlas

- Surge el concepto de **transacción**, como conjunto ordenado de operaciones que, o se ejecutan por completo, o no se ejecutan: La ejecución de una transacción es a todo o nada.
- Si se hiciera una operación y no la otra, generaría un error en la base de datos.

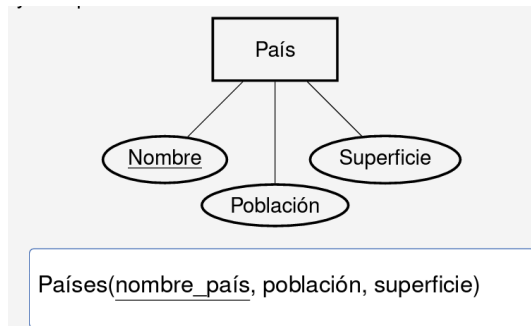
- Si una transacción no puede terminar de realizarse porque una de sus operaciones viola alguna restricción de integridad, entonces debe dejarse la base de datos en el estado anterior al inicio de la misma.

Pasaje de Modelo Conceptual a Modelo Relacional

Podemos hacer el diagrama de Modelo Conceptual y después agarrar eso y pasarlo a un Modelo Relacional. Ese pasaje es automático en general.

En general, cada entidad del modelo ER producirá una relación del modelo relacional (hay excepciones)

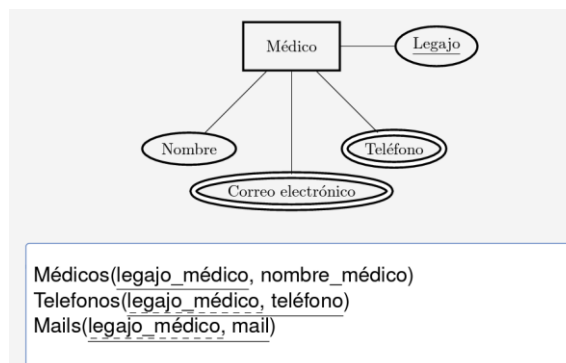
Seria: la Entidad es el nombre de la tabla, y sus atributos son las columnas



Casos Especiales

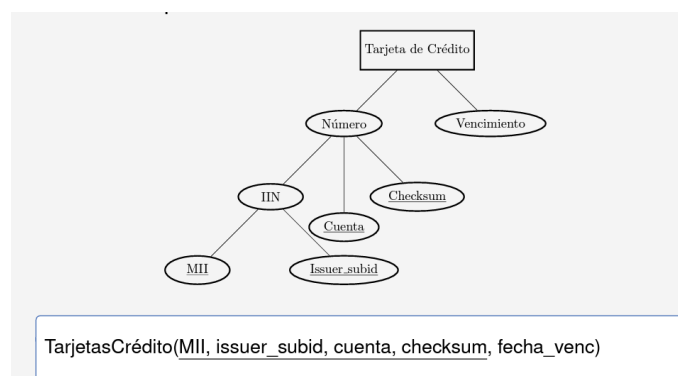
ATRIBUTOS MULTIVALUADOS

Voy a tener que armar otra relación con el Multivaluado, y le agrego lo que identifica a la entidad (acá sería, armo una nueva relación para el Telefono, y agrego el legajo del médico)



ATRIBUTOS COMPUESTOS

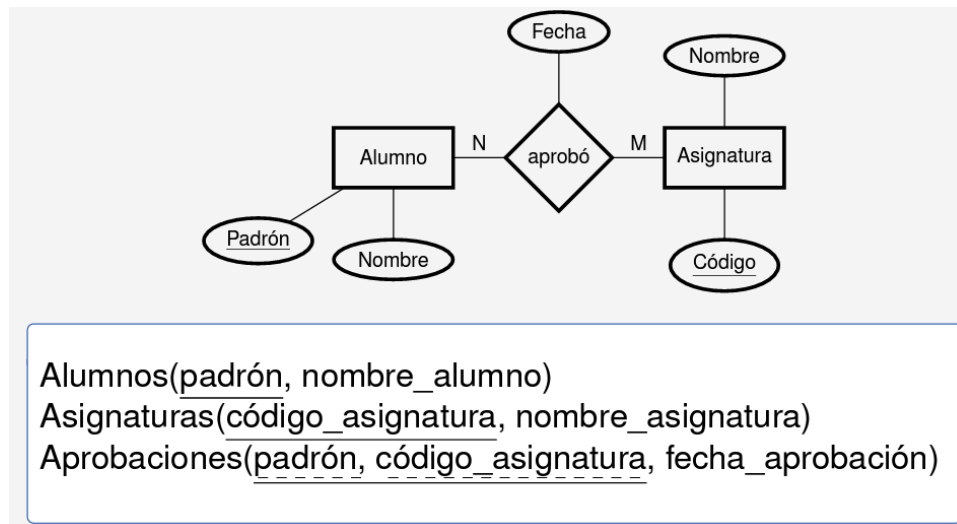
Se representan a través de sus sub-atributos simples (serían como las hojas del árbol)



Representación de las Cardinalidades de las relaciones

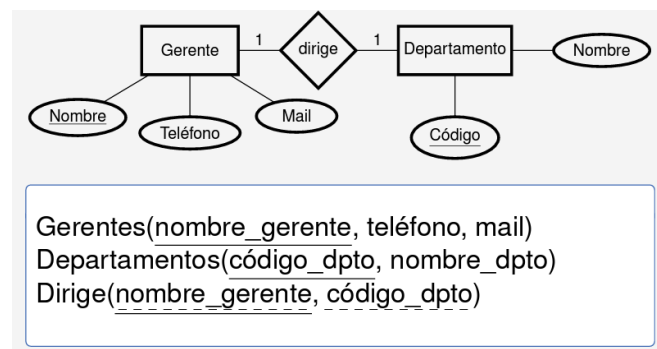
CARDINALIDAD N:M

Cada interrelación N:M del modelo ER producirá una relación del modelo relacional. Como *aprobó* es una relación N:M, ahí tengo una nueva relación del modelo relacional. Se suele llamar tabla de referencias cruzadas



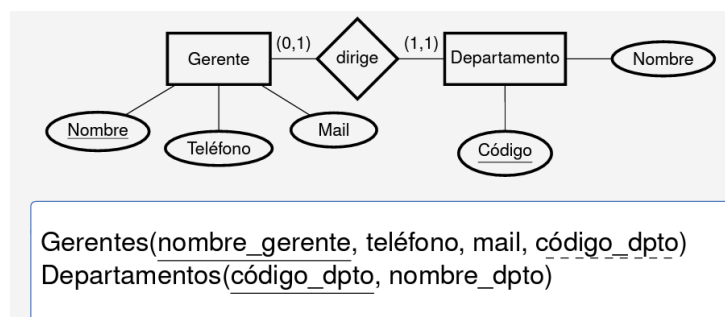
CARDINALIDAD 1:1

Forma 1: relación asociada a la interrelación



Hacemos una relación para la interrelación. Acá, agregaríamos Dirige como nueva relación.

Forma 2.1: recomendada si Gerente tiene participación total



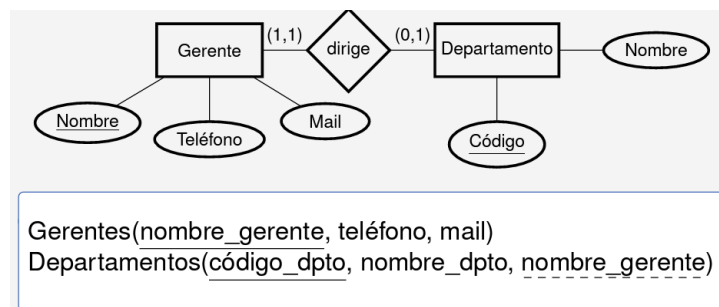
Una interrelación con cardinalidad 1:1 puede representarse incluyendo la clave primaria de una de las entidades participantes como clave foránea en la relación correspondiente a la otra entidad participante, siempre que esta última tenga participación total.

Osea acá el gerente tiene mínimo 1. Acá lo recomendado es agregar el código de depto al Gerente. En vez de agregar una tabla nueva, tengo en Gerentes, una referencia a Departamentos.

Si Dirige hubiera tenido un atributo, lo tengo que poner en Gerente (donde puse la tabla foránea)

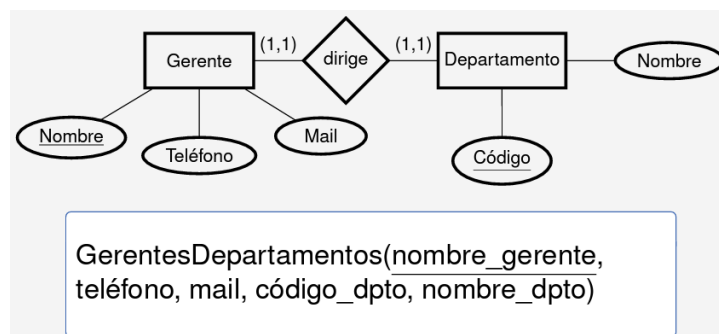
Podría también agregar el nombre de gerente a Departamentos, pero, en esa relacion tengo (0,1): puedo tener un Departamento sin Gerente, entonces tendría valores NULL en esa FK. Usar NULLs no es lo mas recomendable de todo.

Forma 2.2:



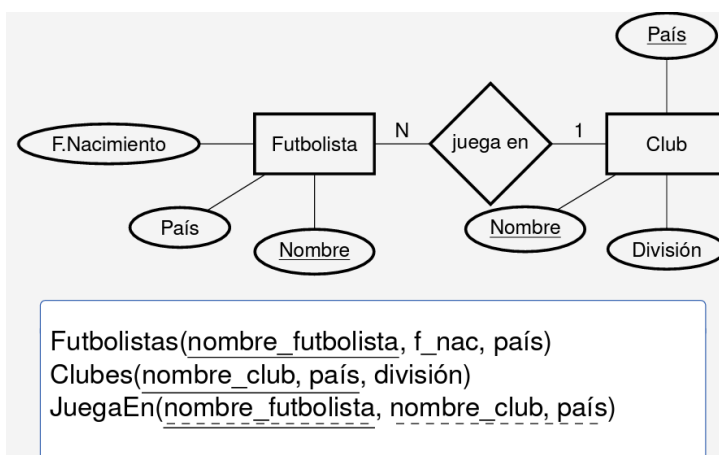
Poner la referencia del lado que tiene la participación total. Acá e slo mismo pero al revés

Forma 3:



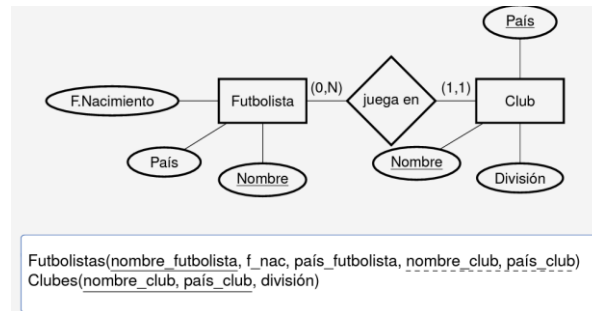
Rebuscada. Está restringida a cuando tengo participación total de ambos lados

CARDINALIDAD 1:N



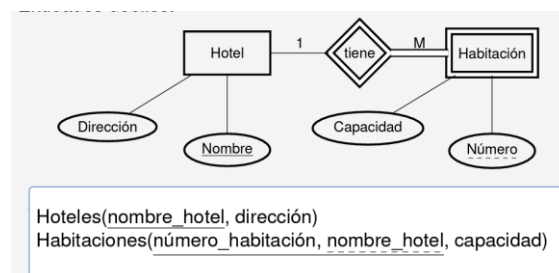
Opción 1: armar una nueva tabla

Si Futbolista tuviera participación total sería conveniente poner una clave foránea directamente.



Entidades débiles

En realidad, es un caso particular de muchos a uno: voy a meter **la clave foránea de la fuerte en la débil**. No tiene sentido agregar una relacion que representa la interrelación “tiene”; es redundante.

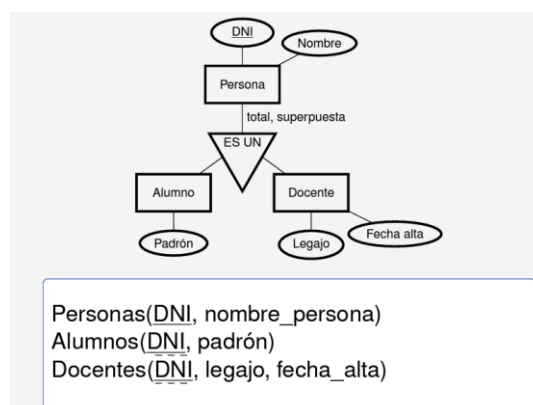


Generalización/Especialización

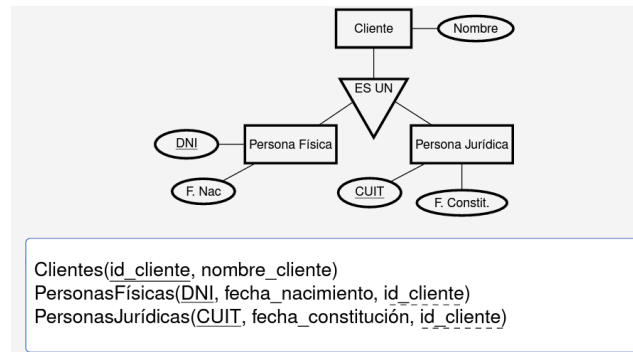
Hay muchas formas de modelar esto. Lo más común es una tabla para cada cosa (una para el supertipo, otra para cada subtipo).

Otra alternativa es armar tabla para los sub tipos y agregarle los atributos del super tipo. En este caso, es redundante.

Si fuera una Generalización parcial, sería incorrecto hacer esto porque no podría representar completamente a los supertipos.



Unión



En este ejemplo debemos crear una **clave subrogada** para identificar a los clientes.

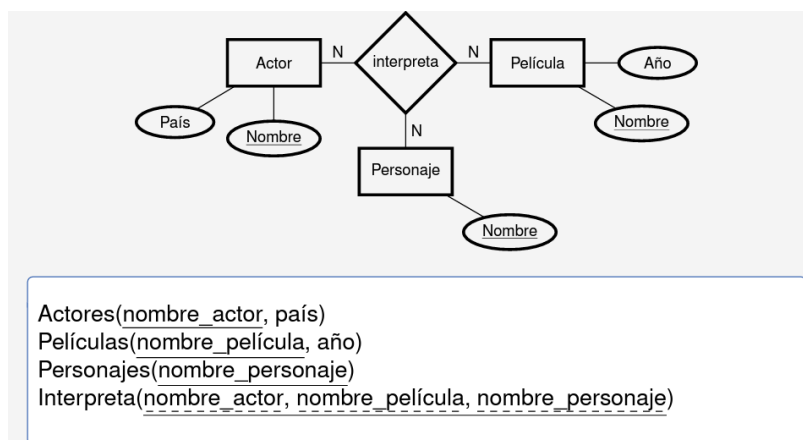
Representación de Interrelaciones Ternarias

Siempre Se crea una relación que representa a la interrelación,

- En esa relación se van a poner las **claves** de los tres tipos de entidades vinculadas
- lo que cambia según la cardinalidad es en cual es la clave primaria de la relación, y cuales son las claves candidatas.

CARDINALIDAD N:N:N

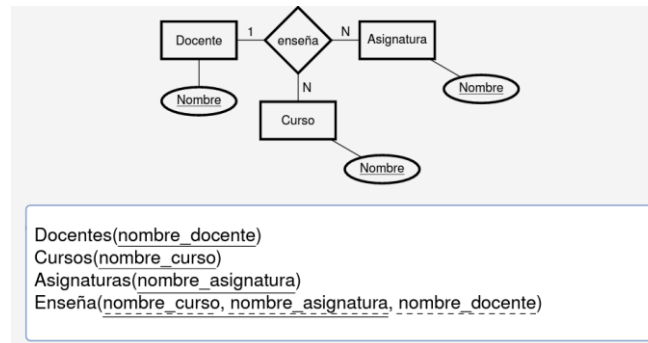
En cada película (Star Wars, ...), distintos actores (Harrison Ford, Carrie Fisher, ...) interpretan distintos personajes (Han Solo, Princesa Leia, ...). A veces un mismo actor puede interpretar más de un personaje en una misma película, e inclusive un mismo personaje en una película puede ser interpretado por más de un actor. Por último, existen personajes que aparecen en muchas películas.



Acá como es N:N:N si o si las 3 claves en Interpreta son claves candidatas

CARDINALIDAD 1:N:N

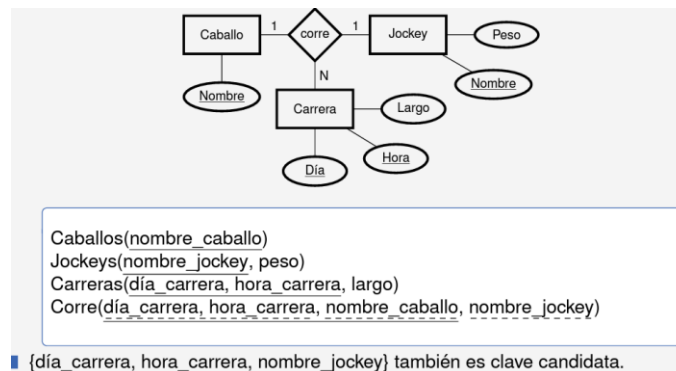
En una escuela, los docentes enseñan distintas asignaturas ("Matemática", "Literatura", ...) en distintos cursos (3°A, 3°B, ...). Cada asignatura en cada curso es enseñada por un único docente.



Aca la clave candidata es nombre_curso, nombre_asignatura porque tengo un 1 en Docente.

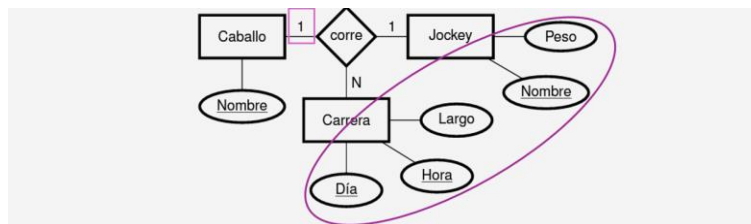
CARDINALIDAD 1:1:N

En un hipódromo se corren varias carreras a diario, en las cuales participan jockeys y caballos. En una carrera, cada jockey está asociado a un caballo, y el caballo sólo es montado por ese único jockey. Sin embargo, en distintas carreras un jockey puede variar de caballo, y un mismo caballo puede ser montado por distintos jockeys.



Voy a tener una clave candidata por cada 1 que tenga. Esa clave candidata va a ser todo el conjunto de atributos de donde sale ese 1:

Opción 1 de clave candidata: {día_carrera, hora_carrera, nombre_jockey}



Opción 2 de clave candidata: {día_carrera, hora_carrera, nombre_caballo}

