

# Diagramas E-R, Relacional y Normalización

BDDA

UT3

Laurent/Rubén



# ¿Qué es?

Representación esquemática de la problemática que debemos plasmar en la BD.

Se divide en varias fases.

- 1º análisis de los requisitos
- 2º modelo lógico
- 3º modelo conceptual y normalización
- 4º paso a tablas (al ordenador)

El análisis de requisitos lo hacen los analistas en contacto con la empresa.

En nuestro caso serán textos con los requisitos.



# ¿Qué es?

**Modelo lógico** estudiaremos el entidad-relación. Más usado para bases de datos SQL.

Es una forma de interpretar el texto y plasmarlo en esquemas. Para ello hay elementos fijos que veremos a continuación.

**Modelo conceptual** es el paso previo a crear la BD. Es una aproximación a lo que serán las tablas partiendo del esquema y requisitos recogidos anteriormente.

Se verá tras el diagrama E-R



# Elementos del diagrama E-R



POLITÉCNICO  
ESTELLA

# Entidades

Son los objetos del texto. Se trata de buscar los sustantivos.

Se representan mediante un rectángulo.

*EJ: Necesitamos crear una Base de Datos que almacene los datos de los alumnos, como su DNI, nombre, apellidos, dirección y teléfono.*

El sustantivo es alumno



Alumno



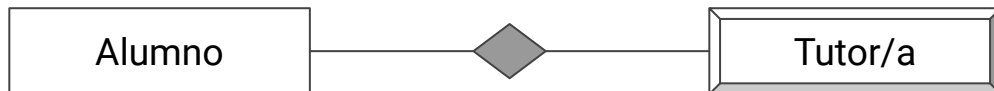
# Entidades

Dos tipos de entidades. Fuertes y débiles.

Fuertes existen solas. Las débiles necesitan de la fuerte para existir. Es decir, si borro la fuerte se elimina la débil.

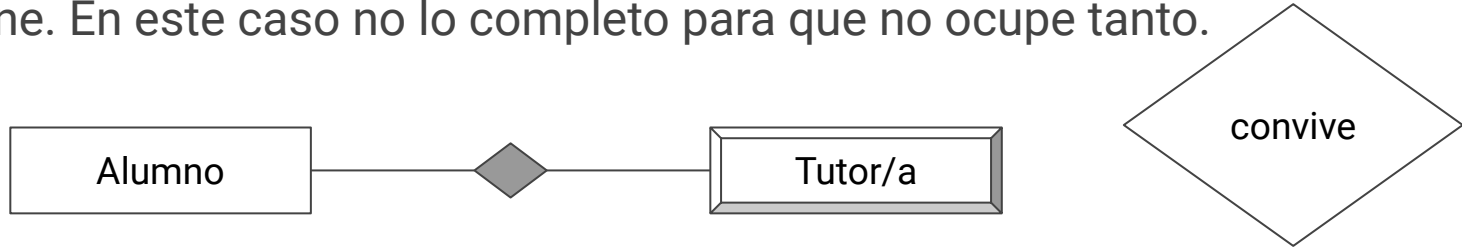
Se representan con un doble rectángulo. Podemos seguir con el ejemplo anterior.

*Los alumnos viven con sus tutor/es legales.* Es decir información de familia o responsables.



# Entidades

El rombo y las líneas nos unen esas relaciones. El rombo debe contener el verbo que les une. En este caso no lo completo para que no ocupe tanto.



En este caso, si borramos los datos del alumno se borrarán los del tutor/a legal. Por ello se llama entidad débil, no sobrevive en la BD sola.

# Atributos

Los campos a guardar de cada entidad. Pueden ser de varios tipos.

- Obligatorios/Opcionales → *Deben existir (ejemplo PK, Not null) vs no hace falta que existan(puede ser Null)*
- Atómicos/compuestos → *Un único elemento vs se puede descomponer en varios atributos a su vez*
- Monovaluado/Multivaluado → *Contiene un valor vs contiene varios*
- Derivados, calculados o almacenados → *El valor proviene de otros cálculos*





# Atributos

Mejor verlo con el ejemplo de alumno. Voy a señalar los atributos del ejercicio.

*Necesitamos crear una Base de Datos que almacene los datos de los alumnos, como su DNI, nombre, apellidos, dirección y teléfono.*

En este ejemplo tenemos todos los tipos de atributos mencionados anteriormente.

Atributos obligatorios los podemos considerar todos menos el teléfono. Si son alumnos pequeños pueden no tenerlo.



# Atributos

DNI, Nombre y Teléfono son atributos simples o atómicos.

Dirección y Apellidos dependerá del diseño, pero deberían ser compuestos:

Apellido → 1º apellido y 2º apellido

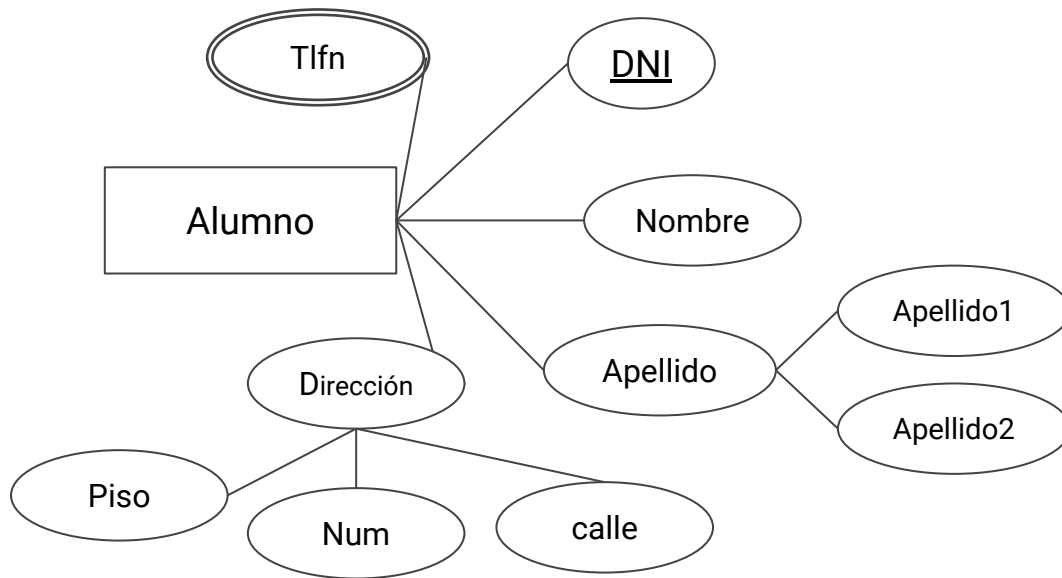
Dirección → Calle, Nº Portal, Piso, Puerta.

Generalmente no se pone en un string toda la información, es más práctico separarlo y tener distintas columnas para búsquedas en nuestra tabla.



# Atributos

Todos los atributos tienen un valor con la excepción del teléfono móvil. Podemos guardar varios teléfonos de un alumno, el fijo y el móvil.



# Entidad y atributos

En el ejemplo anterior hemos visto que es una entidad y como la relacionamos con los atributos.

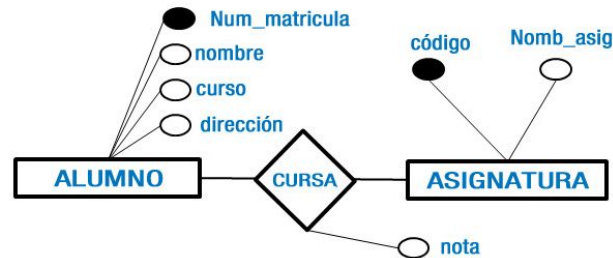
Hay atributos que pueden quedar vacíos, por ejemplo piso. En caso de vivir una vivienda casa no hay número de piso. No se puede indicar todas las restricciones en el diagrama, en esos casos se dejan escritas para plasmarlas en el modelo conceptual



# Atributos de una relación

El ejemplo de los apuntes es muy claro. No solo pueden tener atributos las entidades, sino que la relación que las une, el rombo, puede tener atributos a guardar.

En el ejemplo, no podemos poner nota al alumno o a la asignatura, sino a la unión de ambos. Si indicamos nota en la asignatura no tendría relación con el alumno y viceversa. Por ello, las relaciones tienen atributos. El alumno para esa asignatura tiene una nota concreta.



# Relaciones grado

El grado indica cuantas entidades tienen relación. El concepto más complejo es la relación reflexiva. Consigo mismo.

Por ejemplo, un empleado puede ser responsable y tener a cargo otros empleados. Ese caso es una reflexiva, porque guardamos los mismos tipos de datos. Todos son empleados.

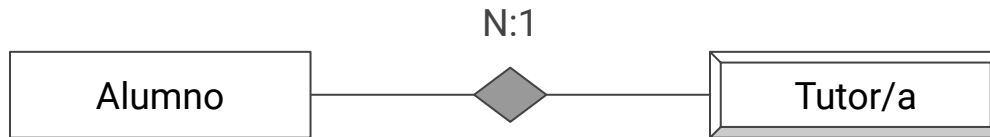
El resto la binaria y ternaria son relaciones entre dos o tres entidades. Más entidades no es habitual.



# Relaciones cardinalidad

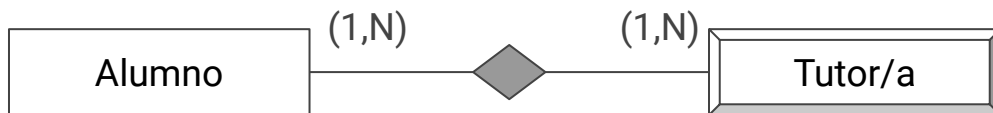
Indica la cantidad que le corresponde a una entidad de la otra entidad.

Ejemplo inicial. Un alumno deberá tener 1 tutor legal o más, obligatoriamente 1. Un tutor legal puede tener 1 o más alumn@s a su cargo, una familia numerosa. Recordemos que son entidad débil, si no hay alumnos no existe la entidad de tutor/a.



# Relaciones cardinalidad

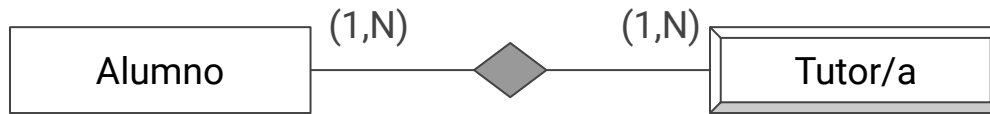
Generalmente se escribe la cardinalidad de entidades. Es muy similar el concepto. En lugar de indicar en el rombo, en la relación, lo indicamos al lado de la entidad. Y se marca el valor mínimo y máximo que puede tener. Los valores son 0, 1 o N. N significa más de 1, no se concreta el número. En nuestro ejemplo quedaría así:





# Relaciones cardinalidad

Como se muestra en los apuntes hay distinta forma de representar en el diagrama la cardinalidad. Dependerá del software a utilizar, en caso de hacerlo a mano lo normal es como en el ejemplo que os indico.



# Software para Diagrama E/R

<https://sourceforge.net/projects/dia-installer/files/latest/download>

Podéis seleccionar otro, este es gratuito y sencillo.



# Software para Modelo relacional

Para el modelo relacional utilicé draw.io que permite almacenar los diagramas en drive directamente.

<https://app.diagrams.net/>

Aquí una explicación de como instalarlo en drive. Aunque se puede utilizar sin realizar estos pasos de forma directa desde el enlace anterior.



# Elementos del diagrama E-R Extendido

# ¿Qué es?

Similar pero permite indicar con mayor exactitud las restricciones.

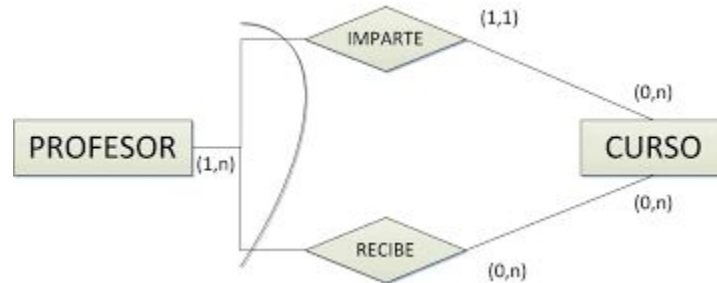
Se añaden nuevos elementos para el diagrama.



# Restricciones en las relaciones

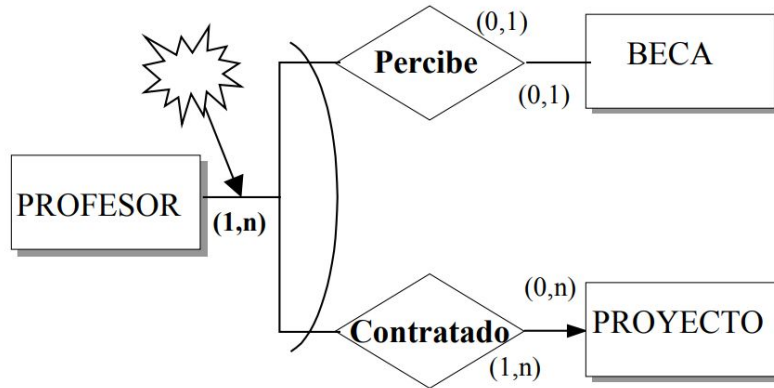
**Exclusividad:** Una entidad puede participar en dos relaciones. Pero solamente puede elegir “un camino”.

Por ejemplo, un profesor o se dedica a impartir clases o a recibirlas, no ambas.



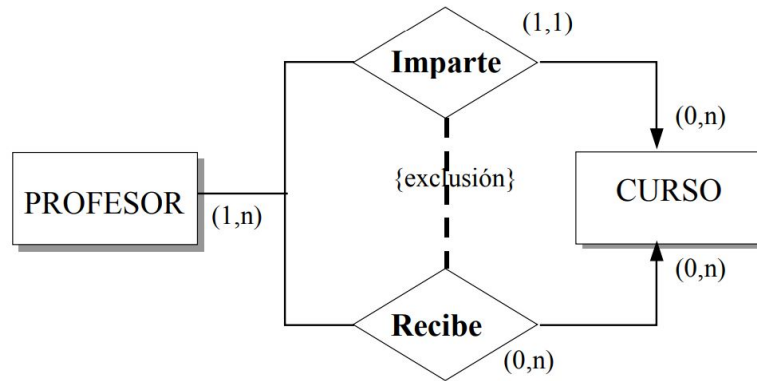
# Restricciones en las relaciones

**Exclusividad:** Otro ejemplo. A un profesor lo contratan o está trabajando con Beca. No puede elegir ambos caminos.



# Restricciones en las relaciones

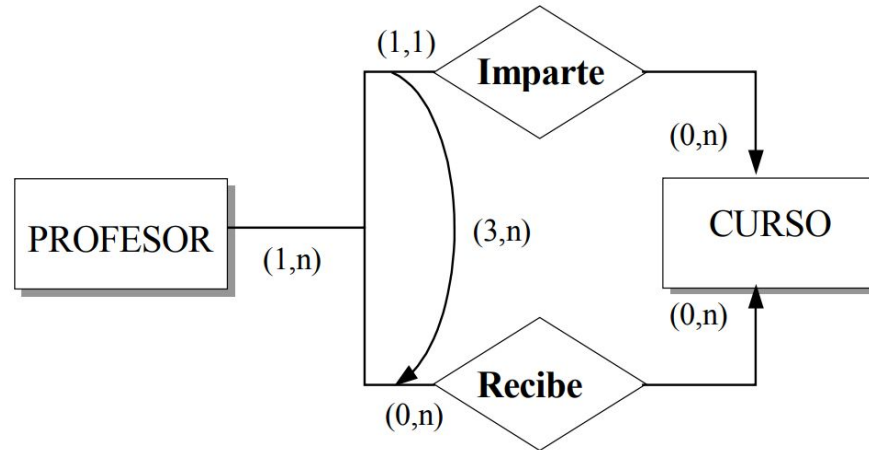
**Exclusión:** Puede ir por ambas relaciones, pero en el caso de ser profesor de una asignatura no puede ser alumno. Este caso muestra un profesor que puede estar matriculado, pero no en las asignaturas que imparte, puede elegir ambos caminos siempre que no coincidan en el mismo curso.





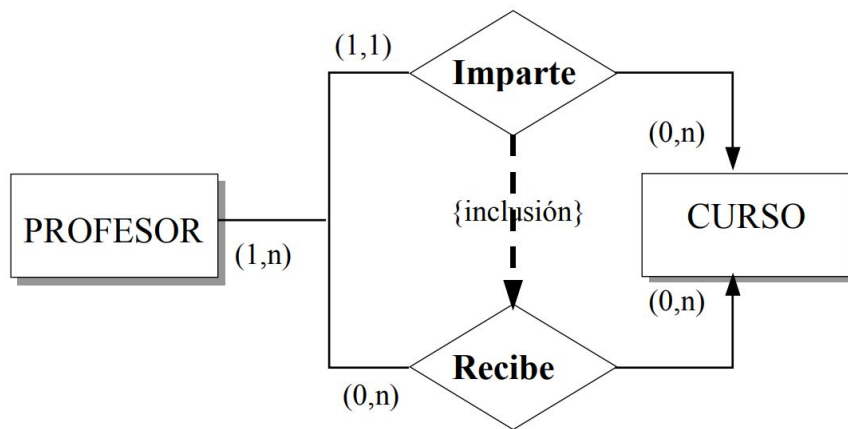
# Restricciones en las relaciones

**Inclusividad:** Implica obligación de ir por un camino de o por el otro con una restricción. Ejemplo, un profesor puede impartir siempre que reciba 3 o más cursos. Vemos que aparecen más números en el modelo extendido, antes solamente 0,1,N.



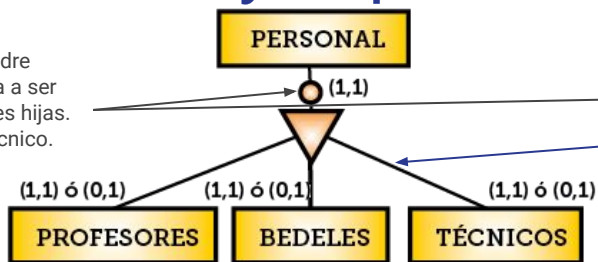
# Restricciones en las relaciones

**Inclusión:** Es obligatorio haber participado en la relación que comparto. Es decir, un profesor puede impartir un curso siempre y cuando lo haya recibido antes.



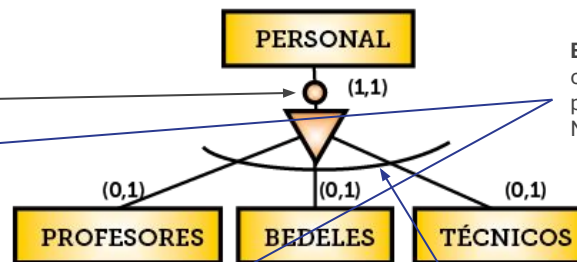
# Generalización y Especialización

**Total:** La entidad padre personal siempre va a ser una de sus entidades hijas. Profesor, bedel o técnico. Se añade el círculo.



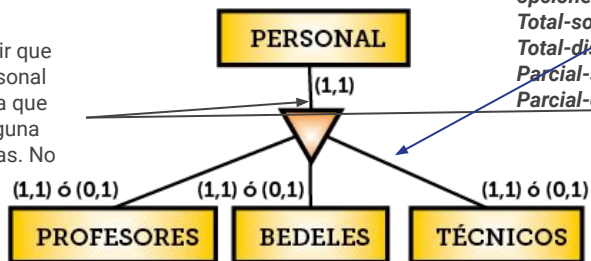
Relación ISA solapada total

**Exclusiva o disjunta:** Una ocurrencia no puede pertenecer a varias hijas. No se añade línea curva



Relación ISA exclusiva total

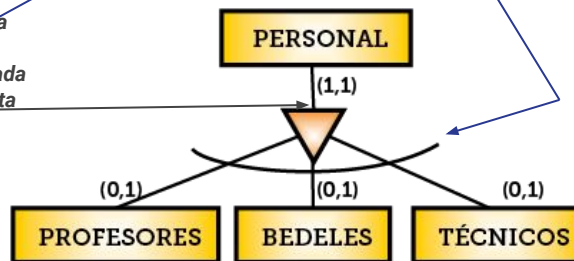
**Parcial:** Puede ocurrir que la entidad padre personal tenga una ocurrencia que no pertenezca a ninguna de sus entidades hijas. No se añade círculo.



Relación ISA solapada parcial

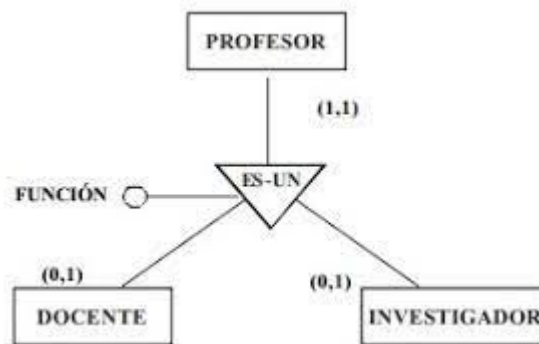
Es posible combinar las 4 opciones:  
Total-solapada  
Total-disjunta  
Parcial-solapada  
Parcial-disjunta

**Parcial:** La herencia puede pertenecer a varias entidades hijas a la vez. Se añade línea curva



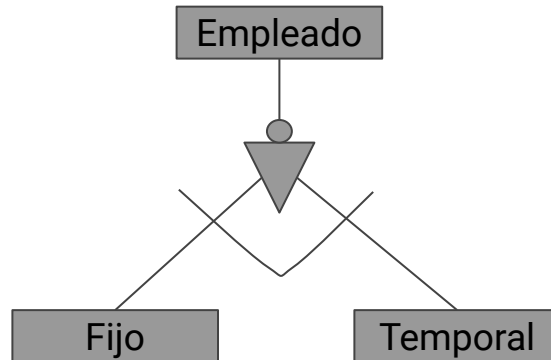
Relación ISA exclusiva parcial

# Generalización y Especialización



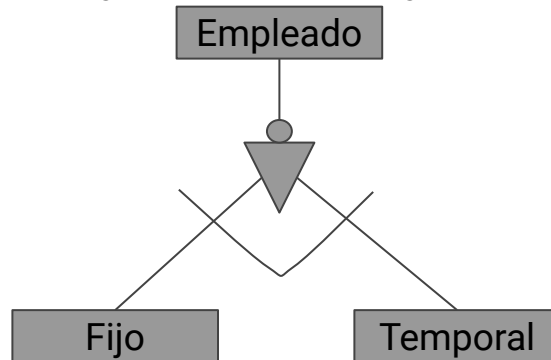
# Generalización y Especialización

**Especialización exclusiva/total:** Solamente puede elegirse una opción. Heredan atributos de la entidad padre y tienen los suyos propios.



# Generalización y Especialización

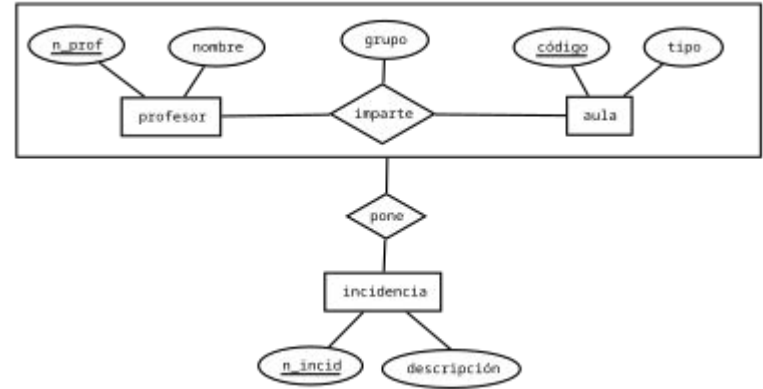
**Especialización exclusiva/total:** Solamente puede elegirse una opción (exclusiva) y todos los empleados se ven reflejados en fijos o temporales. Heredan atributos de la entidad padre y tienen los suyos propios.



# Agregación

Representa relación “parte de”. Permite que una entidad abstracta pueda relacionarse con otra como una relación normal. Las entidades agregadas nunca son débiles ni tienen atributos identificadores, heredan de la relación que las define.

Ejemplo: Profesor imparte una asignatura en el aula. Esa relación es una agregación que se relaciona con la incidencia en el aula.

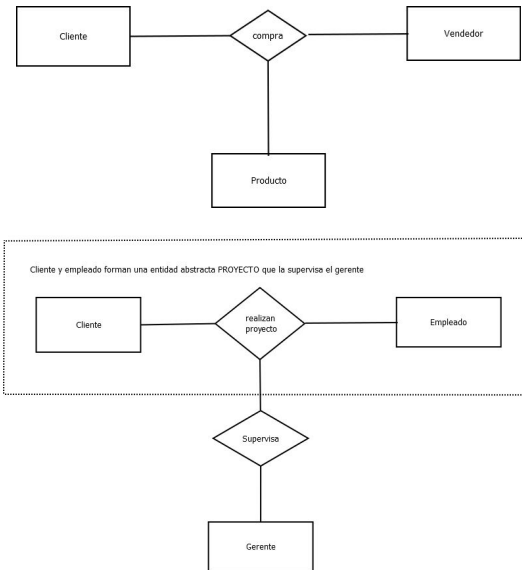


# Agregación (II)

Relación más abstracta que la ternaria. Ternaria las 3 entidades participan de forma simultánea de la relación. En la agregación es más abstracta. Una primera relación binaria, “independiente”, que puede unirse a otra entidad.

Ternaria ej: **Cliente**, **Producto** y **Vendedor**. Una relación ternaria llamada "Compra" podría describir que un **Cliente** compró un **Producto** a través de un **Vendedor**. Aquí, los tres participan en una interacción conjunta.

Agregación ej: Supongamos una relación llamada "Proyectos" que involucra a las entidades **Empleado** y **Cliente**. Ahora, quieres modelar que un **Gerente** supervisa un conjunto de proyectos (es decir, la relación entre **Empleado** y **Cliente**). Aquí, trata la relación "Proyectos" como una entidad para poder conectarla con la entidad **Gerente**.





# MODELO RELACIONAL

# Elementos del modelo

Relación: Estructura más importante. Puede ser una entidad del modelo E-R o una relación entre entidades. Se representa como una TABLA. Los atributos son las columnas de la tabla.

Dominio: Valor que puede tener el atributo. Un tipo de datos o una enumeración por ejemplo.

Atributo: propiedad de la que se guarda información en la tabla acompañada de su dominio

Tupla: Cada fila de la relación, de la tabla.



# Restricciones aceptadas

Clave primaria (PK): un atributo o varias. Representación inequívoca. Única y not null por ser PK

Clave única (Unique): Identificar atributo que no puede tener valores repetidos

Valor No Nulo (VNN): No puede ser NULL

Clave ajena (FK): Identifica un atributo, o conjunto, que es PK en otra tabla. En caso de borrado o actualización se indica:

Borrado/modificación cascada: Se borra de la tabla principal (PK), se borra o modifica en la FK.

Bor/Mod restringida: No deja borrar o modificar la FK.

Bor/Mod a nulos: Si se modifica la PK las FK pasan a valor nulo.

Bor/Mod por defecto: Si se borra la PK las FK pasan a tener valor indicado por defecto.

Restricción de valor (CHECK): condiciones a cumplir por el valor del atributo

Restricciones de integridad (ASSERTIONS): Otras restricciones que no se pueden completar con los anteriores.

# Paso a tablas

**Entidades:** Cada entidad será una tabla.

**VEHÍCULO**(matrícula:dom\_matrícula, marca:dom\_marca,  
modelo:dom\_modelo, cilindrada:dom\_cilindrada, fecha\_fab:fecha)

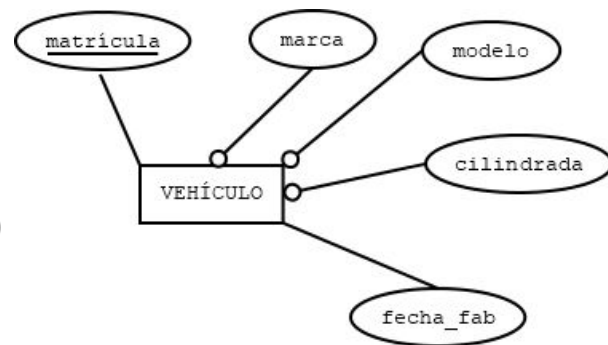
RI:

PK (matrícula)

VNN (modelo)

VNN (marca)

VNN (cilindrada)

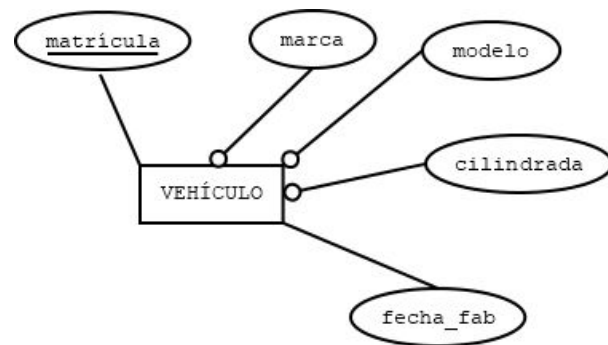


# Paso a tablas

Según los apuntes:

**VEHÍCULO**(matrícula, marca, modelo, cilindrada, fecha\_fab)

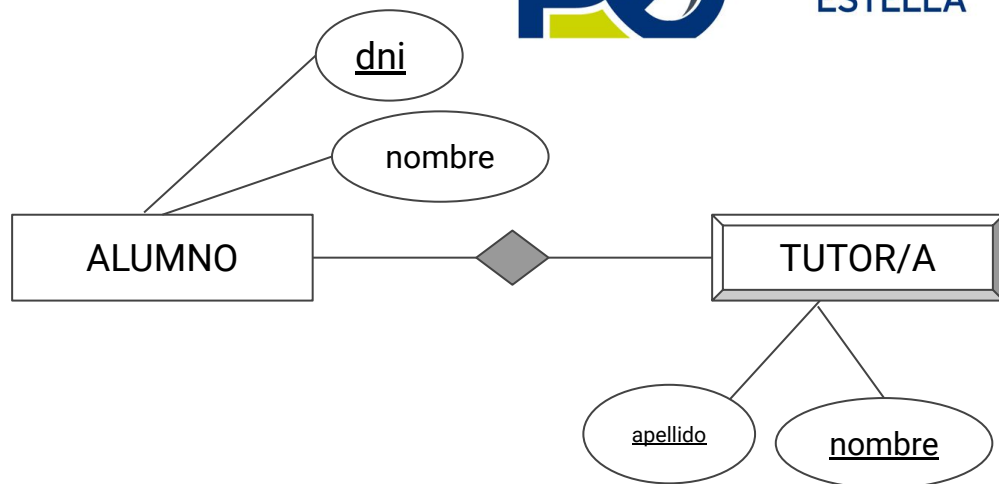
VEHÍCULO	
matrícula	PK
marca	VNN
modelo	VNN
cilindrada	VNN
fecha_fab	



# Paso a tablas



Entidad débil: Absorbe como FK la PK de la entidad fuerte



**ALUMNO**(dni: dom\_dni,  
nombre: dom\_nombre) PK(dni)

**TUTOR/A**(dni: dom\_dni, nombre:  
dom\_nombre, apellido: dom\_apellido)

RI

PK(dni, nombre, apellido)

FK(dni) --> ALUMNO

# Paso a tablas

**Atributos:** Se convierten en campos. No tiene orden definido pero se ponen los que forma la PK primero

- Atributos compuestos se descomponen en simples
- Atributos derivados no se convierten en campos se hará desde la BD
- Atributos multivaluados se convierten en una nueva relación (TABLA)

# Paso a tablas

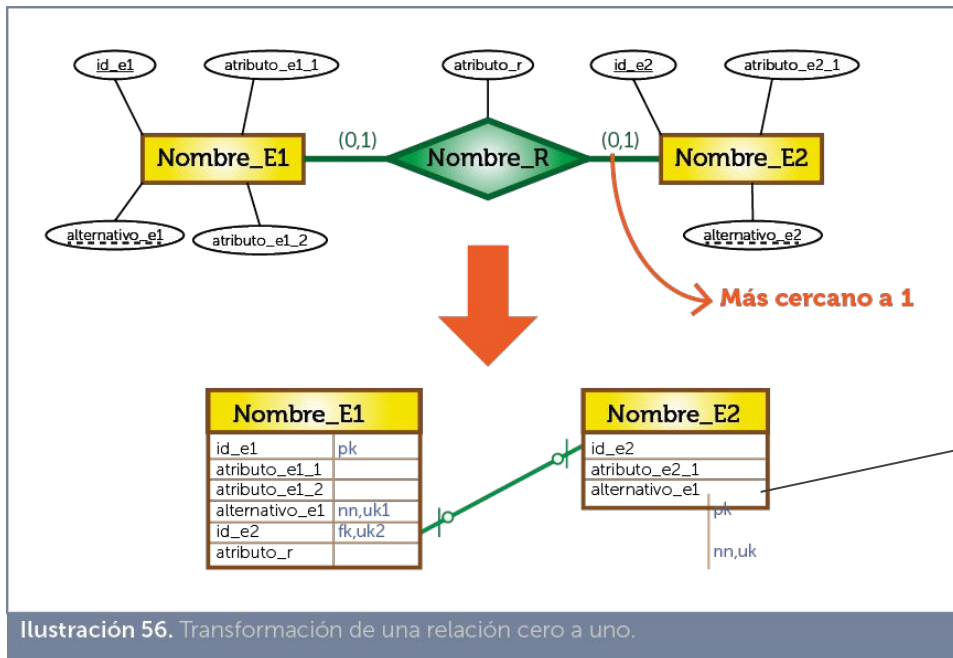
<https://jorgesanchez.net/manuales/gbd/disenio-logico-relacional.html>





# Paso a tablas

## Relaciones 1:1

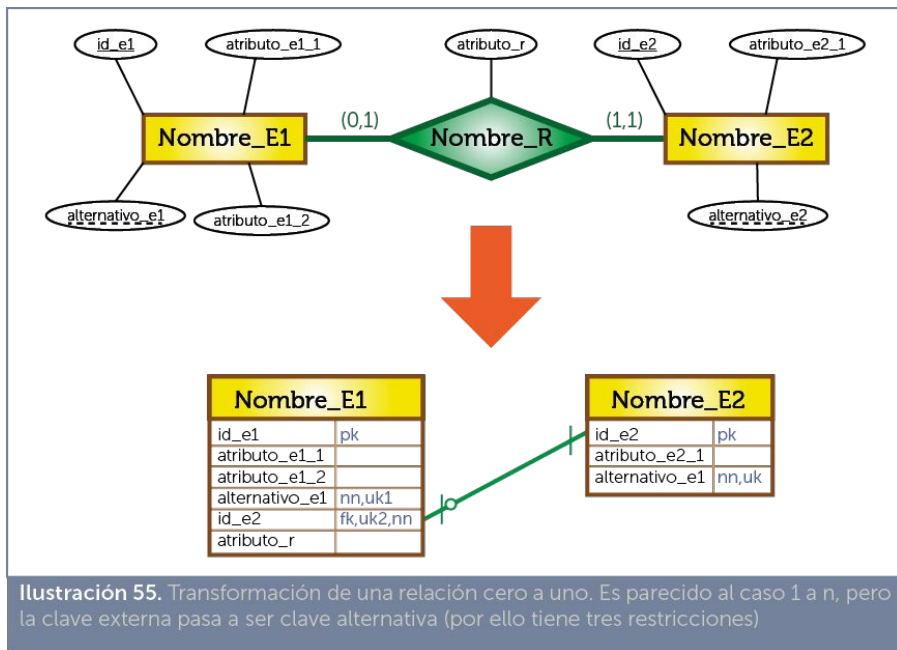


Una tabla absorbe la PK de la otra como FK y el atributo de la relación. Elige el diseñador de la BD.

*En tabla Nombre\_2 atributo alternativo\_e1 es incorrecto y debería ser alternativo\_e2*

# Paso a tablas

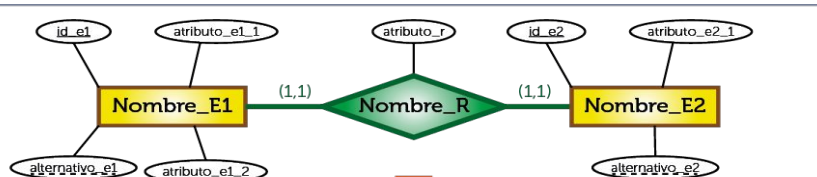
## Relaciones 1:1



La que es (0,1) puede existir o no, por lo tanto absorbe el atributo y la FK.

# Paso a tablas

## Relaciones 1:1



### Solución 1

Nombre_E1	
id_e1	pk
atributo_e1_1	
atributo_e1_2	
alternativo_e1	nn,uk1
atributo_r	
id_e2	uk2,nn
atributo_e2_1	
atributo_e2_2	uk3,nn

### Solución 2

Nombre_E1	
id_e1	pk
atributo_e1_1	
atributo_e1_2	
alternativo_e1	nn,uk1
id_e2	fk,uk2,nn
atributo_r	

Nombre_E2	
id_e2	pk
atributo_e2_1	
atributo_e2_2	
alternativo_e1	nn,uk

### Solución 3

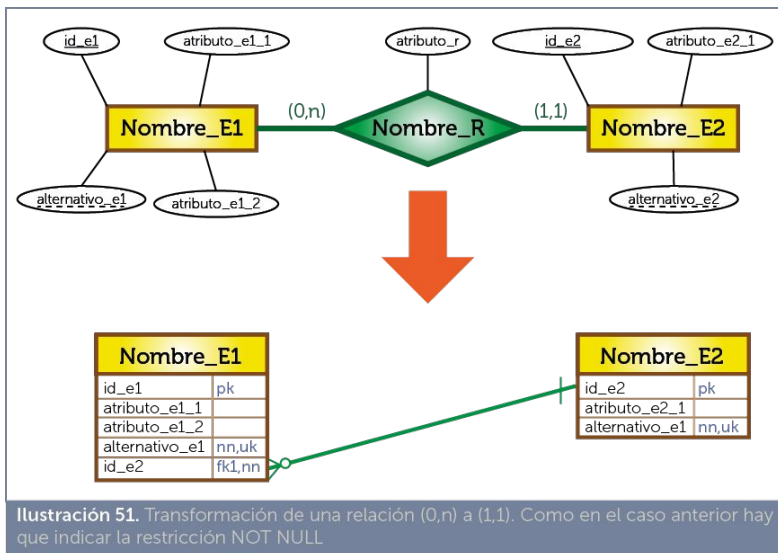
Nombre_E1	
id_e1	pk
atributo_e1_1	
atributo_e1_2	
alternativo_e1	nn,uk1

Nombre_E2	
id_e2	pk
atributo_e2_1	
atributo_e2_2	
alternativo_e1	nn,uk1
atributo_r	
id_e1	nn,fk,uk2

Varias alternativas, la más común, unir todo en una tabla.

# Paso a tablas

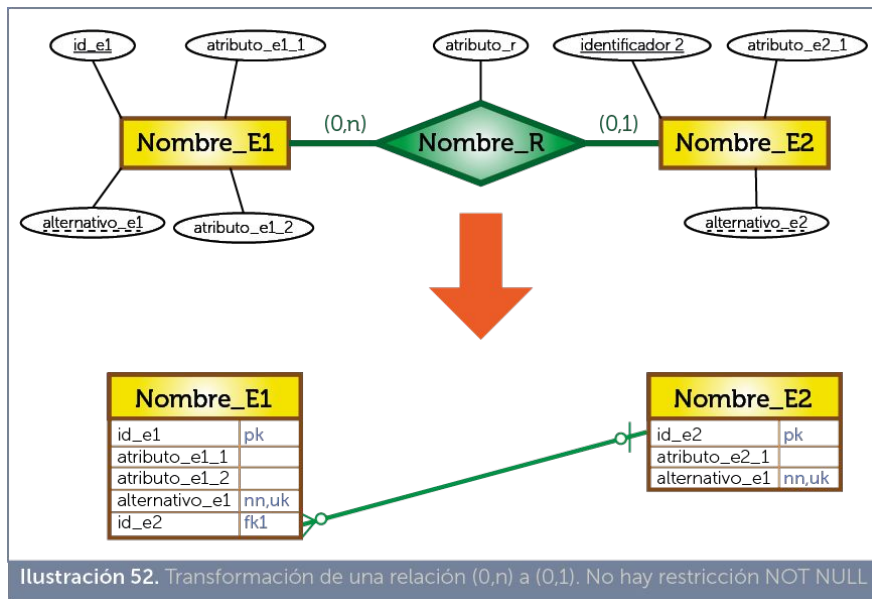
## Relaciones 1:N



La tabla de muchos absorbe el atributo de la relación y la clave ajena apunta a la otra entidad. Se llama propagación de clave y no podrá ser nula

# Paso a tablas

## Relaciones 1:N

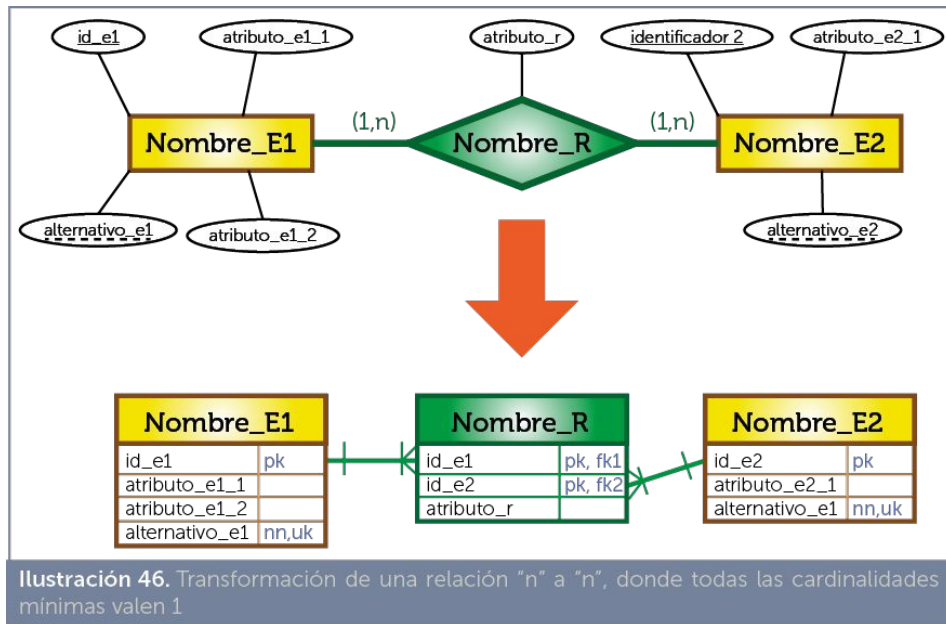


En caso de ser (0,1)  
simplemente indicamos la  
FK y no decimos VNN o NN

Ilustración 52. Transformación de una relación (0,n) a (0,1). No hay restricción NOT NULL

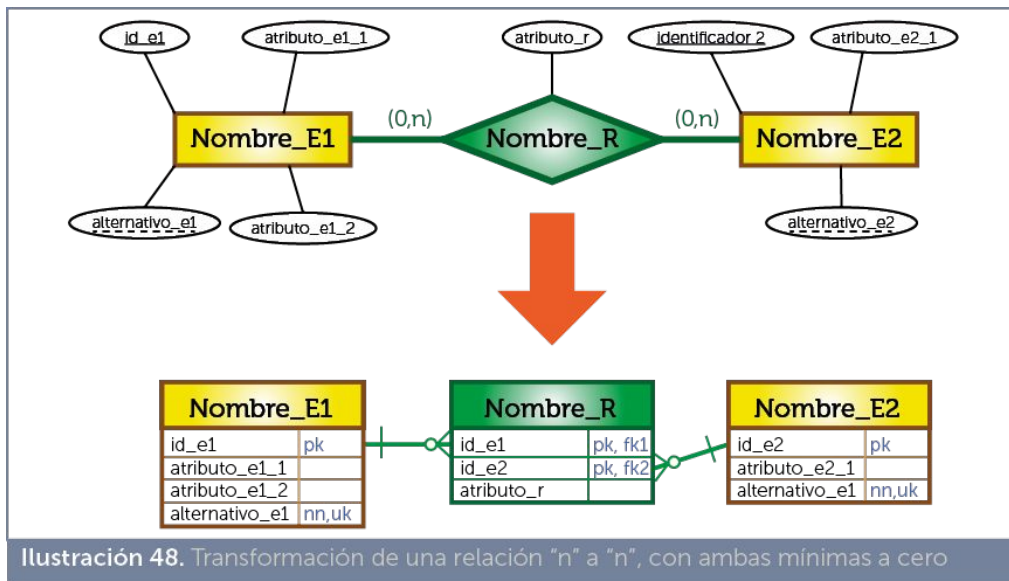
# Paso a tablas

## Relaciones N:M



# Paso a tablas

## Relaciones N:M



# Elaboración de diagramas E-R y paso del diagrama E-R a modelo Relacional



# PDF explicativo

PDF añadido en los recursos del tema.



# Normalización

# ¿Qué es?

Proceso que elimina la redundancia de datos, asegura la integridad y facilita el manejo.

DEPENDENCIAS: Se dice que  $x$  depende de  $y$  si para un único valor de  $y$  se obtiene un único valor de  $x$ .



# 1FN, Primera forma normal

Cada tupla, cada fila, debe tener un valor único y atómico. No pueden existir atributos multivaluados

Cliente			
ID Cliente	Nombre	Apellido	Teléfono
123	Rachel	Ingram	555-861-2025
456	James	Wright	555-403-1659 555-776-4100
789	Cesar	Dure	555-808-9633



Cliente					
ID Cliente	Nombre	Apellido	Teléfono 1	Teléfono 2	Teléfono 3
123	Rachel	Ingram	555-861-2025		
456	James	Wright	555-403-1659	555-776-4100	
789	Cesar	Dure	555-808-9633		

Cliente		
ID Cliente	Nombre	Apellido
123	Rachel	Ingram
456	James	Wright
789	Cesar	Dure



Teléfono del cliente	
ID Cliente	Teléfono
123	555-861-2025
456	555-403-1659
456	555-776-4100
789	555-808-9633

2 Soluciones. Aumentar campos o crear una tabla nueva. Tabla nueva PK compuesta. Mejor solución crear nueva tabla.

## 2FN

Debe cumplir 1FN y, en caso de ser PK compuesta, el resto de atributos dependen de la totalidad de la PK compuesta, no de una parte. Si la PK es un solo atributo no hay que realizar la 2FN.

<b>Título</b>	<b>Formato</b>	Autor	Precio	Editorial
<b>1984</b>	<b>Texto</b>	George Orwell	15	Planeta
<b>1984</b>	<b>Ebook</b>	George Orwell	10	Planeta
<b>El ángel negro</b>	<b>Texto</b>	John Verdon	18	Alguna

Título + formato → precio, pero el resto solo dependen del título



## 2FN

Titulo	Autor	Editorial
1984	George Orwell	Planeta
El ángel negro	John Verdon	Alguna

Titulo	Formato	Precio
1984	Papel	10
1984	Ebook	15
El ángel negro	John Verdon	18

Se divide en dos tablas,  
ahora cada atributo  
depende de la totalidad de  
la clave.



## 2FN, Primera forma normal

Titulo	Autor	Editorial
1984	George Orwell	Planeta
El ángel negro	John Verdon	Alguna

Titulo	Formato	Precio
1984	Papel	10
1984	Ebook	15
El ángel negro	John Verdon	18

Se divide en dos tablas,  
ahora cada atributo  
depende de la totalidad de  
la clave.



# 3FN

Se parte de que 1FN y 2FN se deben cumplir. Los atributos únicamente tendrán dependencia de la clave primaria.

Cod_Alumno	Nombre	Teléfono	Dirección	Código Curso	Nombre Curso	Duración en horas
Al_001	Thor	976111111	Asgard	CUR-01	Estudios Asgardianos	100
Al_002	Tony	976222222	Torre Stark	CUR-02	Tecnología Avanzada	490
Al_003	Bruce	976333333	Dentro de Hulk	CUR-03	Diversión con Rayos Gamma	200

Cod_Alumno	Nombre	Teléfono	Dirección	Código Curso	Código Curso	Nombre Curso	Horas
Al_001	Thor	976111111	Asgard	CUR-01	CUR-01	Estudios Asgardianos	100
Al_002	Tony	976222222	Torre Stark	CUR-02	CUR-02	Tecnología Avanzada	490
Al_003	Bruce	976333333	Interior de Hulk	CUR-03	CUR-03	Diversión con Rayos Gamma	200

[https://www.youtube.com/watch?v=d7-wtp\\_IWGs](https://www.youtube.com/watch?v=d7-wtp_IWGs)

Un ejemplo muy obvio, se separan en tablas.



Está en 3FN y cualquier atributo que tiene dependencia la tiene de la totalidad de la clave primaria. Más restrictiva que 3FN y generalmente es suficiente con 3FN

DNI ¿?	Asignatura ¿?	Tutor ¿?	notas
1	Lenguaje	Eva	5
1	Matemáticas	Andrés	6
3	Lenguaje	Eva	10
2	Matemáticas	Julio	8

Resultado 2 tablas.

Tutorias (DNI, Tutor, notas)  
Asignaturas\_tutor(Asigantura,  
Tutor)

Es un concepto sutil, muy parecido a 3FN, aquí hay varias dos posibles combinaciones de claves candidatas. En caso de usar las 3 como clave 3FN sería correcta.

DNI-Asignatura como clave

DNI-Tutor como clave → la elegimos como clave y será una tabla.

# 4FN

Eliminar dependencias multivaluadas que llevan a redundancia de datos. La 4FN elimina esa redundancia.

Permutaciones de envíos de pizzas

<u>Restaurante</u>	<u>Variedad de Pizza</u>	<u>Área de envío</u>
Vincenzo's Pizza	Corteza gruesa	Springfield
Vincenzo's Pizza	Corteza gruesa	Shelbyville
Vincenzo's Pizza	Corteza fina	Springfield
Vincenzo's Pizza	Corteza fina	Shelbyville
Elite Pizza	Corteza fina	Capital City
Elite Pizza	Corteza rellena	Capital City
A1 Pizza	Corteza gruesa	Springfield
A1 Pizza	Corteza gruesa	Shelbyville
A1 Pizza	Corteza gruesa	Capital City
A1 Pizza	Corteza rellena	Springfield
A1 Pizza	Corteza rellena	Shelbyville
A1 Pizza	Corteza rellena	Capital City



Variedades por restaurante

<u>Restaurante</u>	<u>Variedad de pizza</u>
Vincenzo's Pizza	Corteza gruesa
Vincenzo's Pizza	Corteza fina
Elite Pizza	Corteza fina
Elite Pizza	Corteza rellena
A1 Pizza	Corteza gruesa
A1 Pizza	Corteza rellena

Áreas de envío por restaurante

<u>Restaurante</u>	<u>Área de envío</u>
Vincenzo's Pizza	Springfield
Vincenzo's Pizza	Shelbyville
Elite Pizza	Capital City
A1 Pizza	Springfield
A1 Pizza	Shelbyville
A1 Pizza	Capital City

La primera tabla es correcta que los 3 valores forman un valor único. Pero al ser multivaluada hay redundancia que podemos evitar separando en dos tablas.

# 5FN




Todas las dependencia de join dependen de la clave primaria

- Divide una tabla en tres o más.
- Se puede encontrar en una relación muchos a muchos a muchos.
- FN polémica (depende de la situación).
- Una tabla no está en 5FN si hay una descomposición de esa tabla que muestre la misma información que la original.

	Store	Supplier	Product
▶	West Main Street	J&J Supplies	Large Widget
	West Main Street	J&J Supplies	Small Widget
	West Main Street	Wholesale R Us	Basket
	West Main Street	Wholesale R Us	Bowl
	Hartford Ave #12	J&J Supplies	Large Widget
	Hartford Ave #12	J&J Supplies	Small Widget
	Hartford Ave #12	J&J Supplies	Medium Widget
	Hartford Ave #12	J&J Supplies	Huge Widget
	Hartford Ave #12	J&J Supplies	Non Widget
	Hartford Ave #12	Wholesale R Us	Basket
	Hartford Ave #12	Wholesale R Us	Bowl
	Hartford Ave #12	Wholesale R Us	Container

# 5FN

Todas las dependencias de join dependen de la clave primaria



Store	Product
West Main Street	Large Widget
West Main Street	Small Widget
West Main Street	Basket
West Main Street	Bowl
Hartford Ave #12	Large Widget
Hartford Ave #12	Small Widget
Hartford Ave #12	Medium Widget
Hartford Ave #12	Huge Widget
Hartford Ave #12	Non Widget
Hartford Ave #12	Basket
Hartford Ave #12	Bowl
Hartford Ave #12	Container

Store	Supplier
West Main Street	J&J Supplies
West Main Street	Wholesale R Us
Hartford Ave #12	J&J Supplies
Hartford Ave #12	Wholesale R Us

Supplier	Product
J&J Supplies	Large Widget
J&J Supplies	Small Widget
J&J Supplies	Medium Widget
J&J Supplies	Huge Widget
J&J Supplies	Non Widget
Wholesale R Us	Basket
Wholesale R Us	Bowl
Wholesale R Us	Container

Puedo hacer un join entre las nuevas tablas y obtener la misma información que en la anterior. Menos datos.



FIN UT3