



Invoice

Invoice_id
Customer_id
Order_id
Product_id
Date_time
Status
Total
Remark

UT3. DISEÑO LÓGICO. MODELO RELACIONAL

Módulo: BASES DE DATOS

Curso 2022/2023. 1º DAM

Ruth Lospitao Ruiz



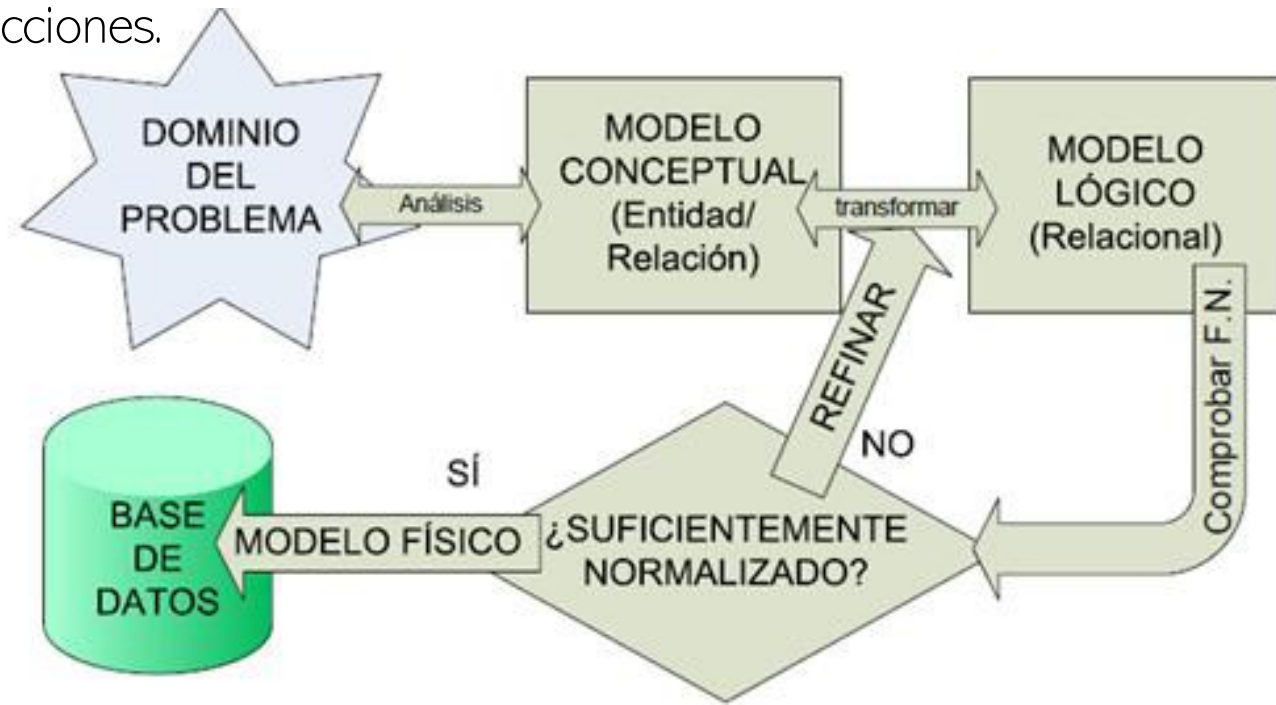
NORMALIZACIÓN: FORMAS NORMALES



INTRODUCCIÓN

La teoría de normalización tiene como fundamento el concepto de formas normales diciendo que una relación está en una determinada forma normal si satisface un cierto conjunto de restricciones.

- Cada forma normal impone una serie de restricciones al conjunto de atributos de un diseño.
- Normalización es conseguir que se cumplan dichas restricciones.
- **Objetivos** de la normalización:
 - Que cosas distintas se almacenen en sitios distintos.
 - Evitar la redundancia.
- El proceso contrario es desnormalizar, y se utiliza para reducir los tiempos de consulta.



OBJETIVOS

- El **objetivo principal del diseño de la base de datos** es generar tablas que modelan los registros en los que se **guardará la información**.
- Es importante que esta información **se almacene sin redundancia** par que se pueda tener una recuperación rápida y eficiente de los datos.
- Es necesario que al diseñar la BD ésta sea **flexible**.
 - La flexibilidad está en el hecho que se puedan agregar datos al sistema posteriormente sin tener que rescribir lo que ya se tiene.
 - Por lo tanto, no tendremos que modificar la estructura de las tablas actuales, simplemente agregar lo que sea necesario
- Es necesario que la BD sea **eficiente**.
 - La eficiencia se refiere al hecho de que no tiene duplicación de datos, y tampoco tienen grandes cantidades de “celdas vacías”.



CONCEPTO

- La normalización es una técnica para diseñar la estructura lógica de los datos de un sistema de información en el modelo relacional, desarrollada por E.F. Codd en 1972
- Es una estrategia de diseño de abajo a arriba: se parte de los atributos y éstos se van agrupando en relaciones (tablas) según su afinidad.
- Es una etapa posterior a la correspondencia entre el esquema conceptual y el esquema lógico, que elimine las dependencias entre atributos no deseadas.



VENTAJAS

- Evita anomalías en inserciones, modificaciones y borrados
- Mejora la independencia de datos
- No establece restricciones artificiales en la estructura de datos.



CONCEPTOS NECESARIOS: DEPENDENCIA FUNCIONAL

- Una dependencia funcional es una relación entre atributos de una misma relación (tabla)
- La dependencia funcional es una noción semántica.
- Si hay o no hay dependencias funcionales entre los atributos lo determinan las reglas de negocio de la organización para la que se diseña la BD
- Cada dependencia funcional es una clase de regla de integridad y representa una relación de uno a muchos.
- Ejemplo: DNI → Nombre y Apellidos



CONCEPTOS NECESARIOS: DEPENDENCIA FUNCIONAL

- Se consideran X, Y, Z atributos

DEPENDENCIA FUNCIONAL REFLEXIVA

- Si "Y" está incluido en "X" entonces $X \rightarrow Y$
- Ejemplo: Si la dirección y nombre están incluidos en DNI entonces con el DNI se puede recuperar la dirección y el nombre

DEPENDENCIA FUNCIONAL AUMENTATIVA

- Si $X \rightarrow Y$ entonces $XZ \rightarrow YZ$
- Ejemplo: DNI \rightarrow nombre

DNI, Dirección \rightarrow Nombre, Dirección

DEPENDENCIA FUNCIONAL TRANSITIVA

- Si $X \rightarrow Y \rightarrow Z$ entonces $X \rightarrow Z$
- Ejemplo: Fecha de nacimiento \rightarrow Edad

Edad \rightarrow Conducir

Fecha de nacimiento \rightarrow Edad \rightarrow Conducir



FORMAS NORMALES

1FN → Codd 1970

2FN → Codd 1970

3FN → Codd 1970

FNBC → Boyce-Codd 1971

4FN → Fagin y Zamiolo 1977

5FN → Rissanen 1978

Recomendable



FORMAS NORMALES: 1FN

- Una tabla está en Primera Forma Normal (1FN o FN1) si, y solo si, todos los atributos de la misma contienen valores atómicos, es decir, no hay grupos repetitivos.
- Dicho de otra forma, estará en 1FN si los atributos no clave, dependen funcionalmente de la clave.
- Por lo general la mayoría de las relaciones cumplen con estas características, así que podemos decir que la mayoría de las relaciones se encuentran en 1FN



¿CÓMO SE NORMALIZA A 1FN?

1. Se crea, a partir de una tabla inicial, una nueva tabla cuyos atributos son los que presentan dependencia funcional de la clave primaria. La clave de esta será la misma clave primaria de la tabla inicial. Esta tabla ya estará en 1FN
2. Con los atributos restantes se crea otra tabla y se elige entre ellos uno que será la clave primaria de dicha tabla. Comprobaremos si esta segunda tabla está en 1FN.
 - a) Si es está en 1FN, la tabla inicial ya está en 1FN y el proceso termina
 - b) Si no está en 1FN, tomaremos la segunda tabla como la tabla inicial y repetimos el proceso

NOTA: Si hemos evitado el uso de atributos compuestos y multivaluados habremos llegado a este punto encontrándonos en 1FN



¿CÓMO SE NORMALIZA A 1FN?

EJEMPLO

Suponemos que nos encontramos con una entidad ALUMNO que el atributo teléfonos es multivaluado porque puede tener varios valores.

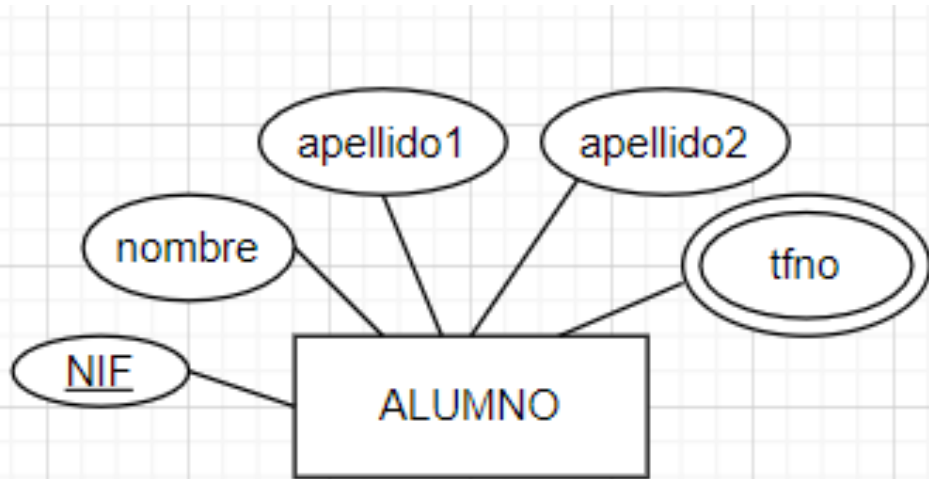
ALUMNO (NIF, Nombre, Apellido1, Apellido2, Tfno*)

- a) Dejamos la tabla con los atributos clave mas los atributos clave que dependan funcionalmente de la clave, es decir, en este caso los que no son multivaluados.

ALUMNO1 (NIF, Nombre, Apellido1, Apellido2)

- b) Hacemos otra tabla con los atributos clave mas el atributo que no cumpla la dependencia funcional (el multivaluado) y se amplía la clave. Luego se renombra nueva tabla

ALUMNO2 (NIF, Tfno) → TELEFONO (NIF, Tfno)



FORMAS NORMALES: 1FN

- Solución: Si un atributo tiene varios valores, cree una entidad adicional y relaciónala con la entidad original mediante una relación 1:N

NIF (PK)	NOMBRE	APELLIDO1	APELLIDO2	TELEFONO
45678B	Miguel	Pérez	Casas	915456854
56487C	María	Martín	López	915486456 914584568
54867D	Eva	Rodriguez	García	914789456



DNI (PK)	NOMBRE	APELLIDO1	APELLIDO2
45678B	Miguel	Pérez	Casas
56487C	María	Martín	López
54867D	Eva	Rodriguez	García

DNI (PK FK)	TELEFONO (PK)
45678B	915456854
56487C	915486456
56487C	914584568
54867D	914789456



FORMAS NORMALES: 2FN

- Una relación está en Segunda Forma Normal (2FN) si y solo si está en 1FN y todos los atributos no clave tiene dependencia funcional completa de la clave primaria.
- De acuerdo a esta definición, cada tabla que tiene un atributo único como clave está en 2FN



¿CÓMO SE NORMALIZA A 2FN?

NOTA: Se hace hincapié en que esto solo será necesario si la clave primaria está formada por varios atributos. Si la clave primaria es un solo atributo no hacemos nada porque ya estaría en 2FN

1. Se crea, a partir de la tabla inicial, una nueva tabla con los atributos que dependen funcionalmente de forma completa de la clave. La clave de esta tabla será la misma clave primaria de la tabla inicial. Esta tabla ya estará en 2FN.
2. Con el resto de atributos, se crea otra tabla que tendrá por clave el subconjunto de atributos de la clave inicial de los que dependen de forma completa. Se comprueba si esta tabla está en 2FN.
 - a) Si está en 2FN, la tabla inicial ya está normalizada y el proceso termina.
 - b) Si no está en 2FN, tomamos esa segunda tabla como tabla inicial y repetimos el proceso



¿CÓMO SE NORMALIZA A 2FN?

Si tuviera un único campo PK
ya estaría en 2FN

EJEMPLO: Supongamos que tenemos la relación RESERVA de la siguiente manera donde un cliente ha reservado un determinado coche para su alquiler:

RESERVA (DNI_conductor, Matricula, Nombre_conductor, Fecha_permiso, Código_Marca_Coche, Marca_Coche, Modelo, Color)

- a) Haríamos una tabla con los atributos clave y los atributos no primos (cuando decimos atributos no primos nos referimos a los que no pertenecen a la clave principal) que cumplan la dependencia funcional completa

RESERVA1 (DNI_conductor, Matricula) → renombramos a RESERVA (DNI_conductor, Matricula)

- b) Haríamos una tabla por cada subconjunto de atributos clave y atributos no primos que dependan de cada uno de ellos.

RESERVA2 (DNI_Conductor, Nombre_conductor, Fecha_permiso)

→ renombramos a CONDUCTOR (DNI_Conductor, Nombre_conductor, Fecha_permiso)

RESERVA3 (Matricula, Codigo_Marca_Coche, Marca_Coche, Modelo, Color)

→ renombramos VEHÍCULO (Matricula, Codigo_Marca_Coche, Marca_Coche, Modelo, Color)



FORMAS NORMALES: 2FN

EJEMPLO2: Modelos de vehículos donde se almacena el fabricante, modelo, año y país del fabricante

- Cualquier atributo que no sea parte de la clave primaria (PK) debe depender de toda la PK, y no solo de algún atributo que compone la PK
- Solución: Crear una tabla adicional con la PK parcial

MODELOS_DE_VEHICULOS
FABRICANTE (PK)
MODELO (PK)
ANIO_DE_LANZAMIENTO
PAIS_DEL_FABRICANTE



MODELOS_DE_VEHICULOS
FABRICANTE (PK FK)
MODELO (PK)
ANIO_DE_LANZAMIENTO

FABRICANTES
FABRICANTE (PK)
PAIS_DEL_FABRICANTE



FORMAS NORMALES: 3FN

- Una relación está en 3FN si y solo si está en 2FN y todos sus atributos no clave dependen no transitivamente de la clave primaria.
- Dicho de otro modo, si y solo si los atributos no clave son mutuamente independientes y son dependientes por completo de la clave primaria



¿CÓMO SE NORMALIZA A 3FN?

1. Se crea, a partir de la tabla inicial, una nueva tabla con los atributos que no poseen dependencias transitivas de la clave primaria. Esta tabla ya estará en 3FN
2. Con los atributos restantes, se crea otra tabla con los dos atributos no clave que intervienen en la dependencia transitiva, y se elige uno de ellos como clave primaria, si cumple los requisitos para ello. Se comprueba si esta tabla está en 3FN
 - a) Si está en 3FN la tabla ya está normalizada y el proceso termina
 - b) Si no está en 3FN tomamos esta segunda tabla como tabla inicial y repetimos proceso



¿CÓMO SE NORMALIZA A 3FN?

EJEMPLO: Retomando el ejemplo de RESERVA de coches para su alquiler.

RESERVA (DNI_conductor, Matricula)

CONDUCTOR ((DNI_Conductor, Nombre_conductor, Fecha_permiso)

VEHÍCULO (Matricula, Código_Marca_Coche, Marca_Coche, Modelo, Color)

Observamos que RESERVA y CONDUCTOR ya están en 3FN porque no hay dependencias funcionales transitivas, pero en VEHÍCULO sí encontramos una dependencia funcional transitiva.

Matrícula → Código_marca_coche → Marca_coche

Por tanto procederemos de la siguiente forma:



¿CÓMO SE NORMALIZA A 3FN?

EJEMPLO: Retomando el ejemplo de RESERVA de coches para su alquiler.

RESERVA (DNI_conductor, Matricula)

CONDUCTOR ((DNI_Conductor, Nombre_conductor, Fecha_permiso)

VEHÍCULO (Matricula, Código_Marca_Coche, Marca_Coche, Modelo, Color)

- a) Dejamos la tabla con su atributo clave y los atributos no primos que no tienen dependencia funcional transitiva.

VEHÍCULO1 (Matrícula, Codigo_Marca_Coche, Modelo, Color)

→ VEHÍCULO (Matrícula, Codigo_Marca_Coche, Modelo, Color)

- b) Creamos otra tabla con los atributos no primos que tienen dependencia funcional transitiva y ponemos como clave el atributo no primo que aparece en la tabla anterior

VEHÍCULO2 (Codigo_Marca_Coche, Marca_Coche) → renombramos MARCA (Codigo_Marca_Coche, Marca_Coche)



FORMAS NORMALES: 3FN

- EJEMPLO2: Información de la tabla vehículos que se muestra en la imagen
- Ningún atributo que no sea parte de la PK puede depender de otro atributo que no sea PK. Es decir, **no puede haber dependencias transitivas**.
- Solución: Mover el atributo a una nueva entidad.

VEHICULOS
MATRICULA (PK)
COLOR
DNI_PROPIETARIO
PAIS_PROPIETARIO

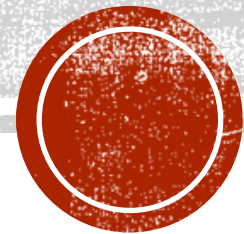


VEHICULOS
MATRICULA (PK)
COLOR
DNI_PROPIETARIO (FK)

PROPIETARIOS
DNI_PROPIETARIO (PK)
PAIS_PROPIETARIO



GRAFO RELACIONAL



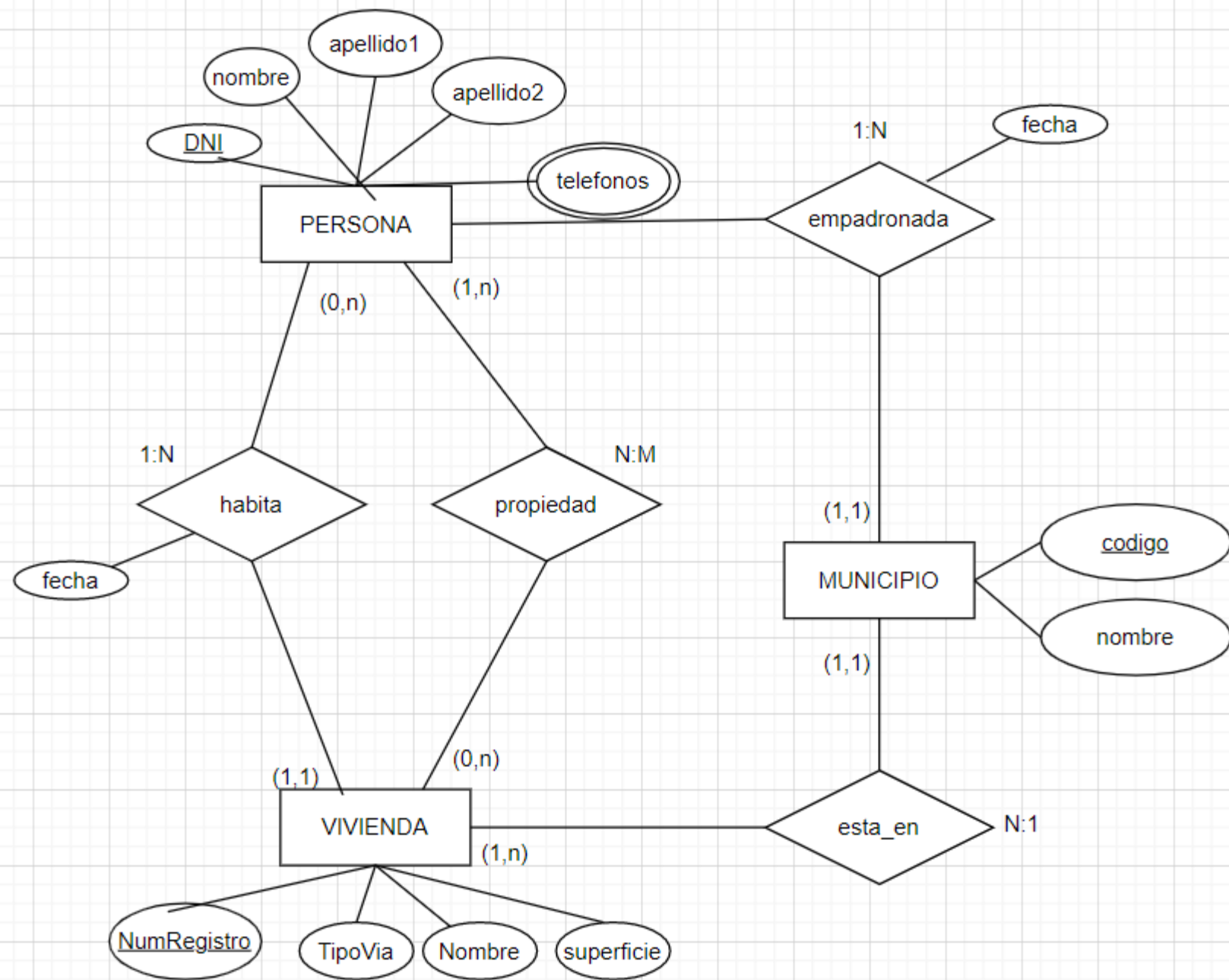
CONCEPTO

- Grafo relacional: Forma de representar gráficamente el esquema relacional de una manera sencilla y completa
- Es un grafo compuesto de un conjunto de nodos, donde cada nodo representa un esquema de la relación, es decir, una tabla de la BD.
- Para cada tabla, como mínimo, ha de aparecer su nombre y atributos, indicando su clave primaria (subrayando)
- Además, se dibujan una serie de arcos que conectan los atributos que constituyen la clave ajena con la tabla referenciada, permitiendo así que el usuario entienda los campos clave que comparte dominios comunes.
- En definitiva, los arcos representan la referenciabilidad de los atributos (clave ajena) de una relación respecto a la clave primaria de la otra.
- Los arcos están direccionados de modo que el arco parta de la clave ajena y la flecha señale a la tabla referenciada.



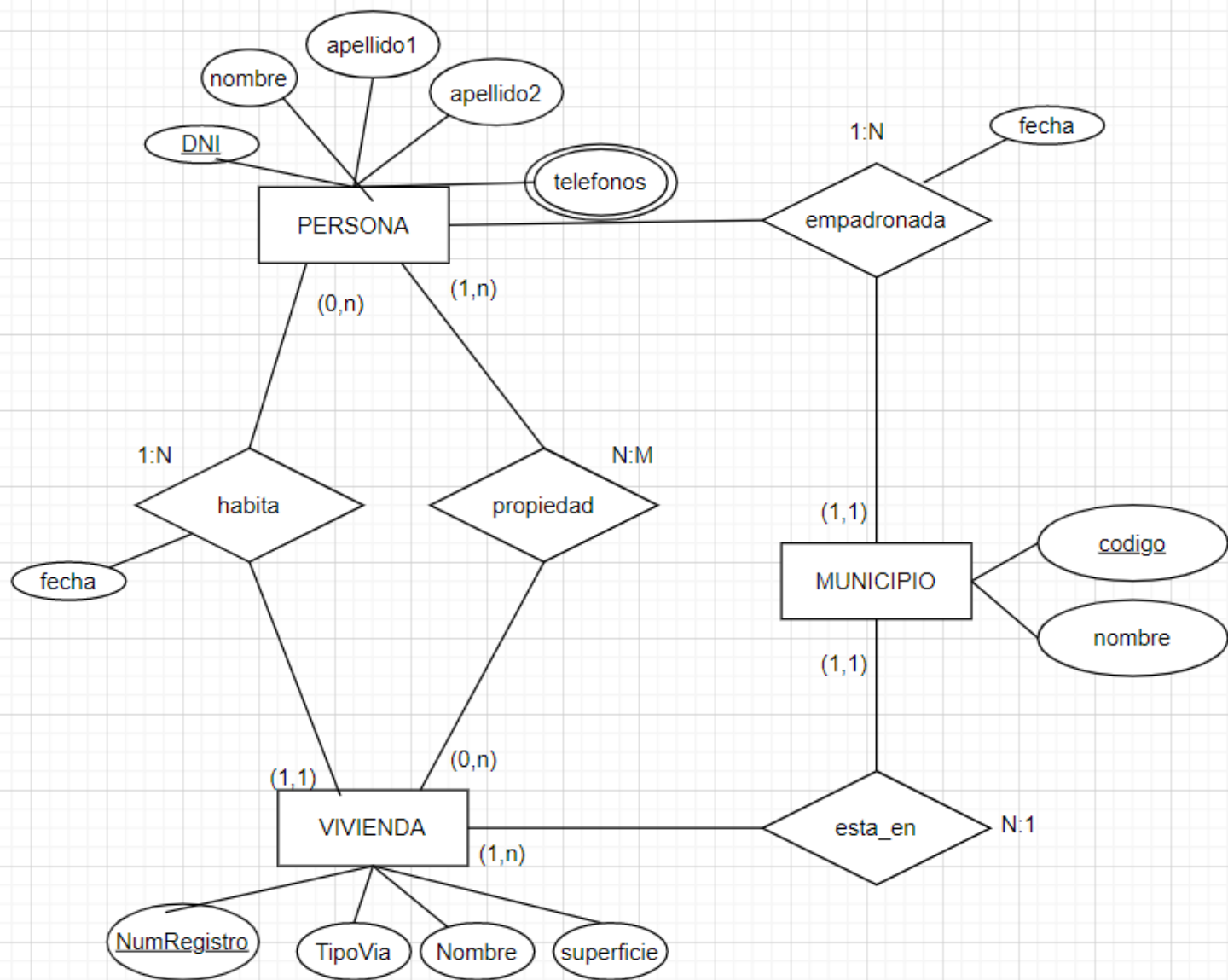
EJEMPLO

- Modelar la relación entre personas, viviendas y municipios.
- Cada persona solo puede estar empadronada en un único municipio, y habitar en una vivienda, pero puede ser propietaria de varios inmuebles. Si diagrama e/R sería:



EJEMPLO

- TELEFONOS (DNI, teléfono)
- PERSONA (DNI, Nombre, Apellido1, Apellido2)
- VIVIENDA (NumRegistro, TipoVia, nombre, superficie)
- MUNICIPIO (Código, nombre)
- EMPADRONADA (DNI, código, fecha)
- HABITA (DNI, NumRegistro, Fecha)
- PROPIEDAD (DNI, NumRegistro)



EJEMPLO

TELEFONOS (DNI, teléfono)

PERSONA (DNI, Nombre, Apellido1, Apellido2)

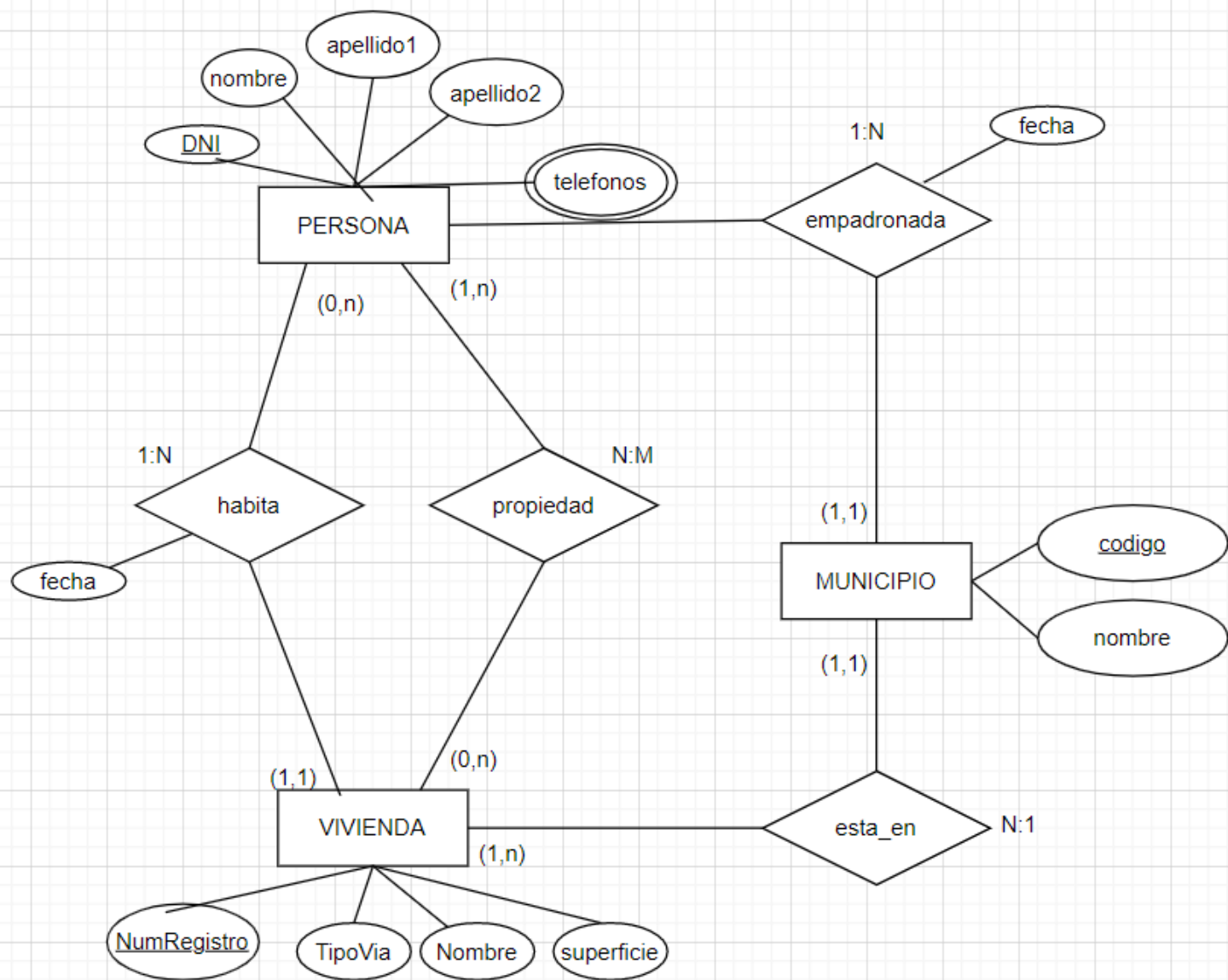
VIVIENDA (NumRegistro, TipoVia, nombre, superficie)

MUNICIPIO (Código, nombre)

EMPADRONADA (DNI, código, fecha)

HABITA (DNI, NumRegistro, Fecha)

PROPIEDAD (DNI, NumRegistro)





Invoice

Invoice_id
Customer_id
Order_id
Product_id
Date_time
Status
Total
Remark

UT3. DISEÑO LÓGICO. MODELO RELACIONAL

Módulo: BASES DE DATOS

Curso 2022/2023. 1º DAM

Ruth Lospitao Ruiz

