

ÉRICA PALOMINO

TEMA 3

MODELO RELACIONAL

erica.palomino@escuelaartegranada.com

ESCUELAARTEGRANADA

EL MODELO RELACIONAL

Una **base de datos relacional** es un conjunto de una o más tablas.

Las tablas se organizan en filas y columnas

Para todas las filas se debe cumplir que: **dos filas con la clave principal igual deben tener el resto de atributos igual.**

Las tablas se relacionan entre sí por un campo en común.



PASO DEL MODELO E-R AL MODELO RELACIONAL

PASO A TABLAS

En el modelo Entidad/Relación hay una serie de elementos que sirven para representar la lógica del problema planteado.

Todos los elementos deben aparecer representados en el modelo relacional.



ENTIDADES

ENTIDADES

Cada entidad crea obligatoriamente una tabla.

- **Nombre de la tabla:** nombre de la entidad
- **Columnas de la tabla:** atributos de la entidad
- **Clave principal:** clave primaria de la entidad



RELACIONES

RELACIONES

El caso de las relaciones hay que estudiarlo una a una. Se debe tener en cuenta la cardinalidad de la relación para poder actuar en consecuencia.

Las relaciones en general, todas crean una tabla nueva.

- **Nombre de tabla:** Nombre de la relación
- **Columnas de la tabla:** Todos los atributos que pueda tener
- **Clave principal:** la unión de las claves principales de las entidades que une
 - Hay que estudiar si los atributos de la propia relación deben formar parte de la clave o no

RELACIONES CARDINALIDAD 1:N

Las relaciones con cardinalidad 1:N crean una tabla nueva como el resto:

- **Nombre tabla:** Nombre de la relación
- **Columnas:** Todos los atributos que pueda tener
- **Clave principal:** la unión de las claves principales de las entidades que une

RELACIONES CARDINALIDAD 1:N

Las relaciones con cardinalidad 1:N crean una tabla nueva como el resto:

- **Nombre tabla:** Nombre de la relación
- **Columnas:** Todos los atributos que pueda tener
- **Clave principal:** la unión de las claves principales de las entidades que une

En este tipo de relaciones nos podemos ahorrar la tabla añadiendo (**propagando**) los valores hacia la tabla cuyos registros solo se relacionan con un registro de la otra tabla.

RELACIONES CARDINALIDAD 1:1

Estas relaciones son un caso especial de las relaciones 1:N.

Para saber si una relación 1:1 desaparece o se mantiene hay que estudiar la **participación**.

Se tomará siempre la decisión que deje menos **huecos en blanco**.

RELACIONES CARDINALIDAD 1:1

Ambas participaciones (0,1): la relación se mantiene como si fuera del tipo N:N

(1,1) – (0,1): se propaga hacia la entidad que tiene un 0 en la participación y la relación desaparece.

Ambas participaciones (1,1): se puede propagar en cualquier dirección y la relación desaparece

ENTIDADES DÉBILES

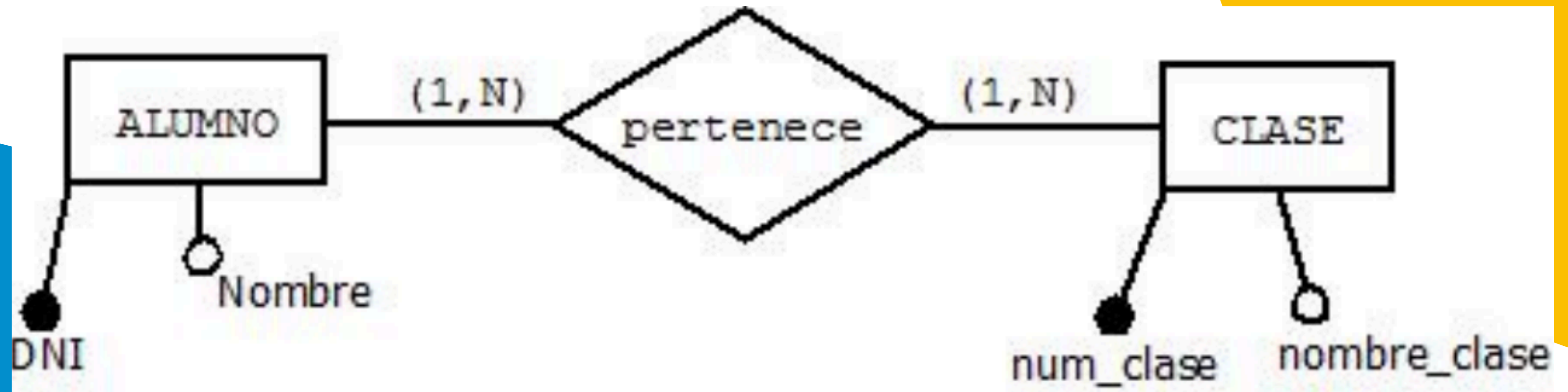
Cada entidad débil genera una nueva tabla del modelo relacional.

- **Nombre de la tabla:** Nombre de la entidad débil
- **Columnas de la tabla:** Atributos de la entidad débil
- **Clave principal:** Unión de la clave de la entidad débil con la clave de su entidad fuerte.

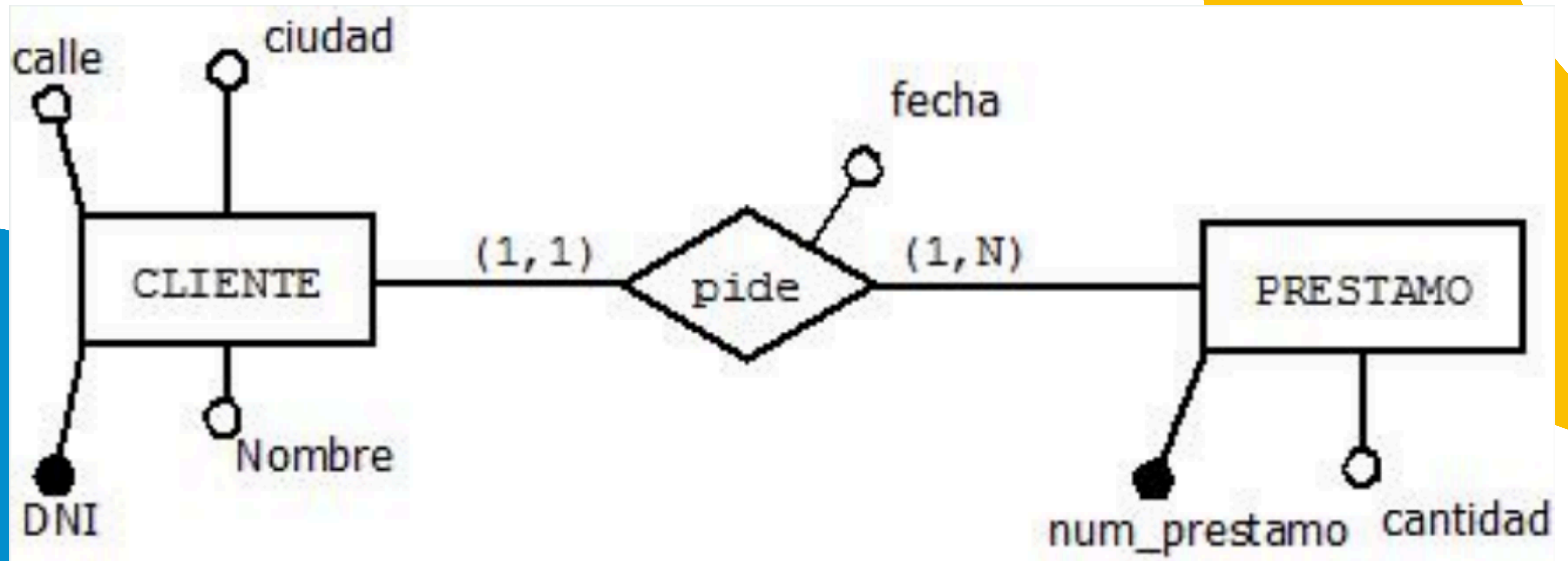


EJERCICIOS

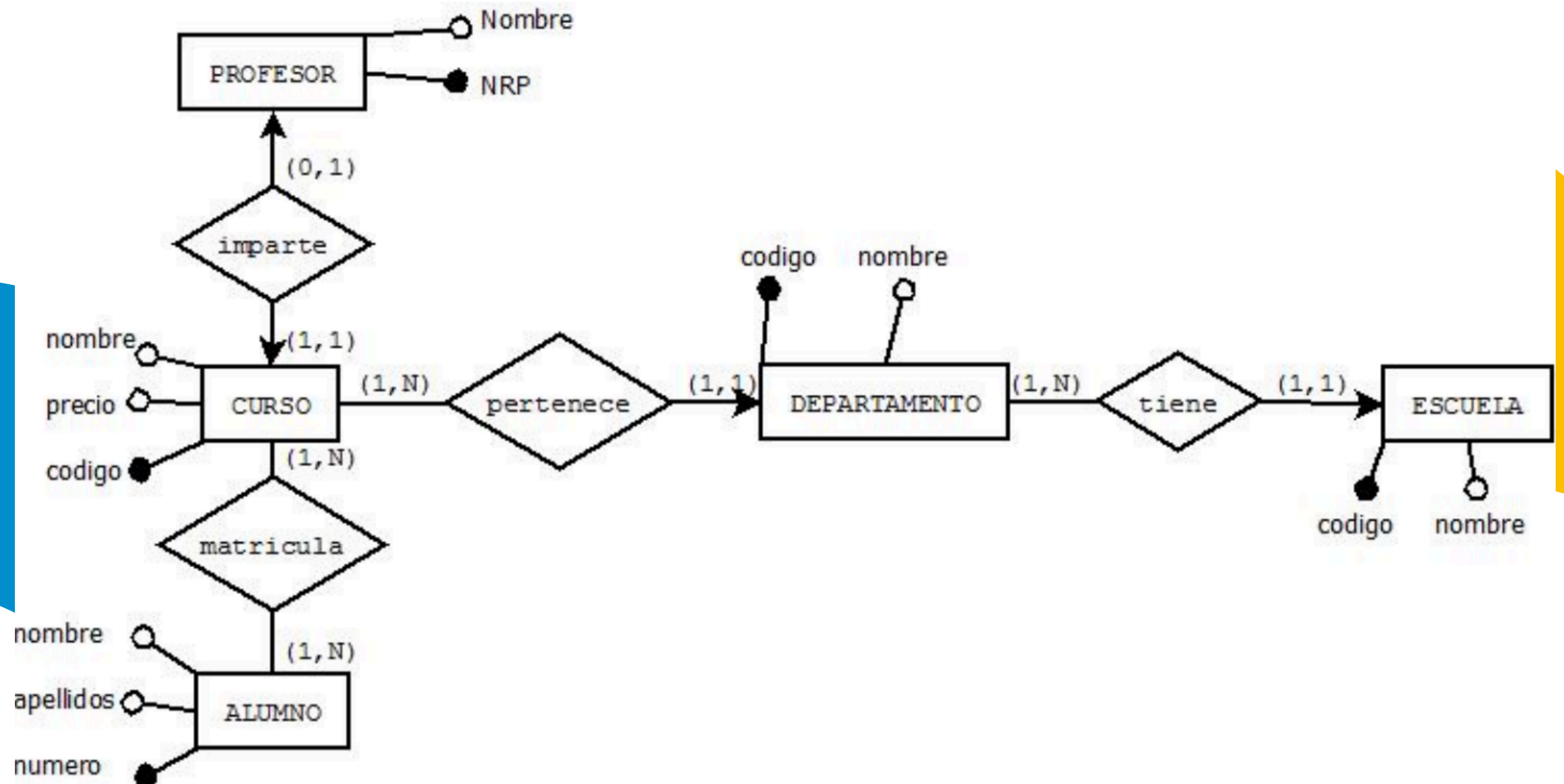
EJERCICIOS



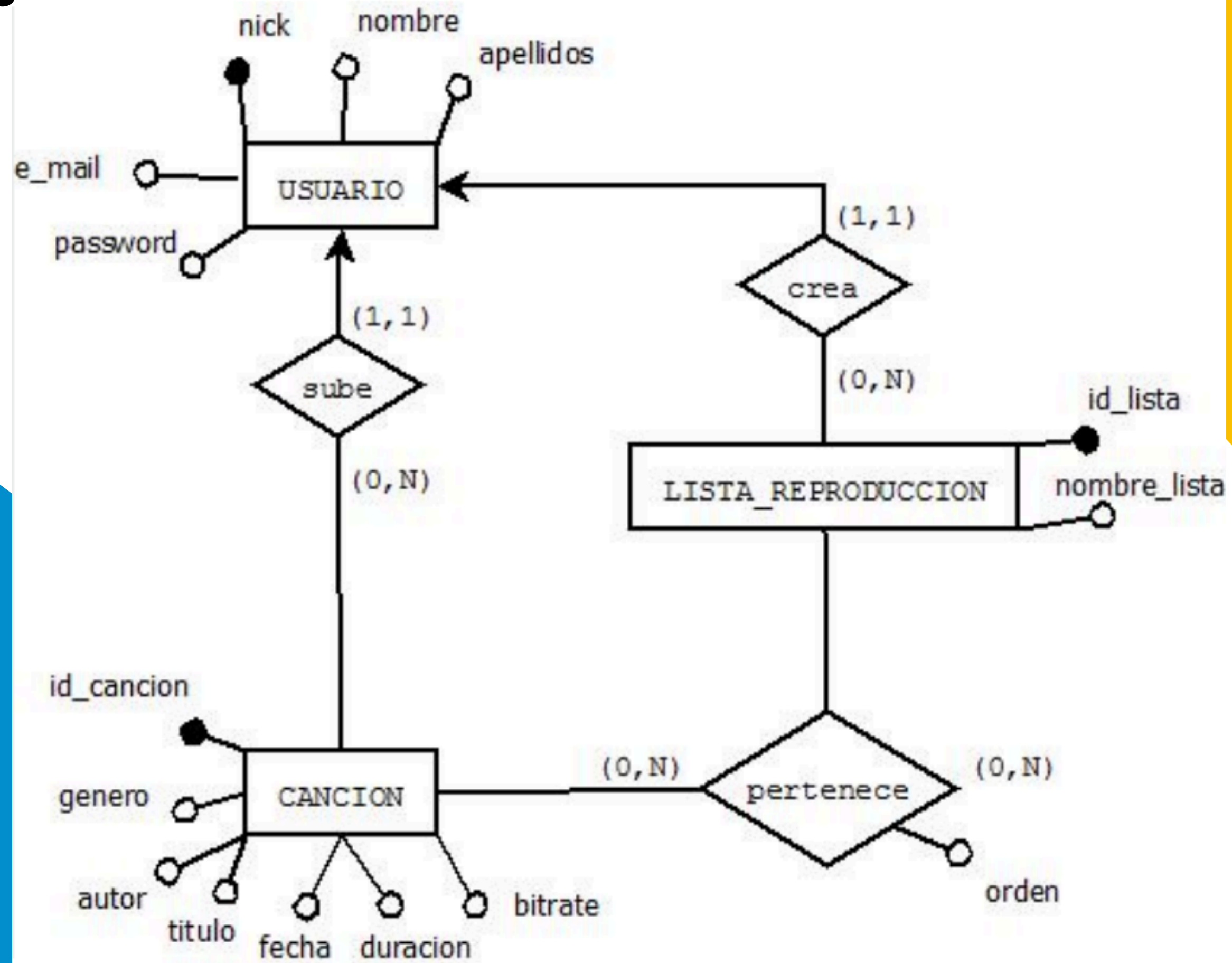
EJERCICIOS



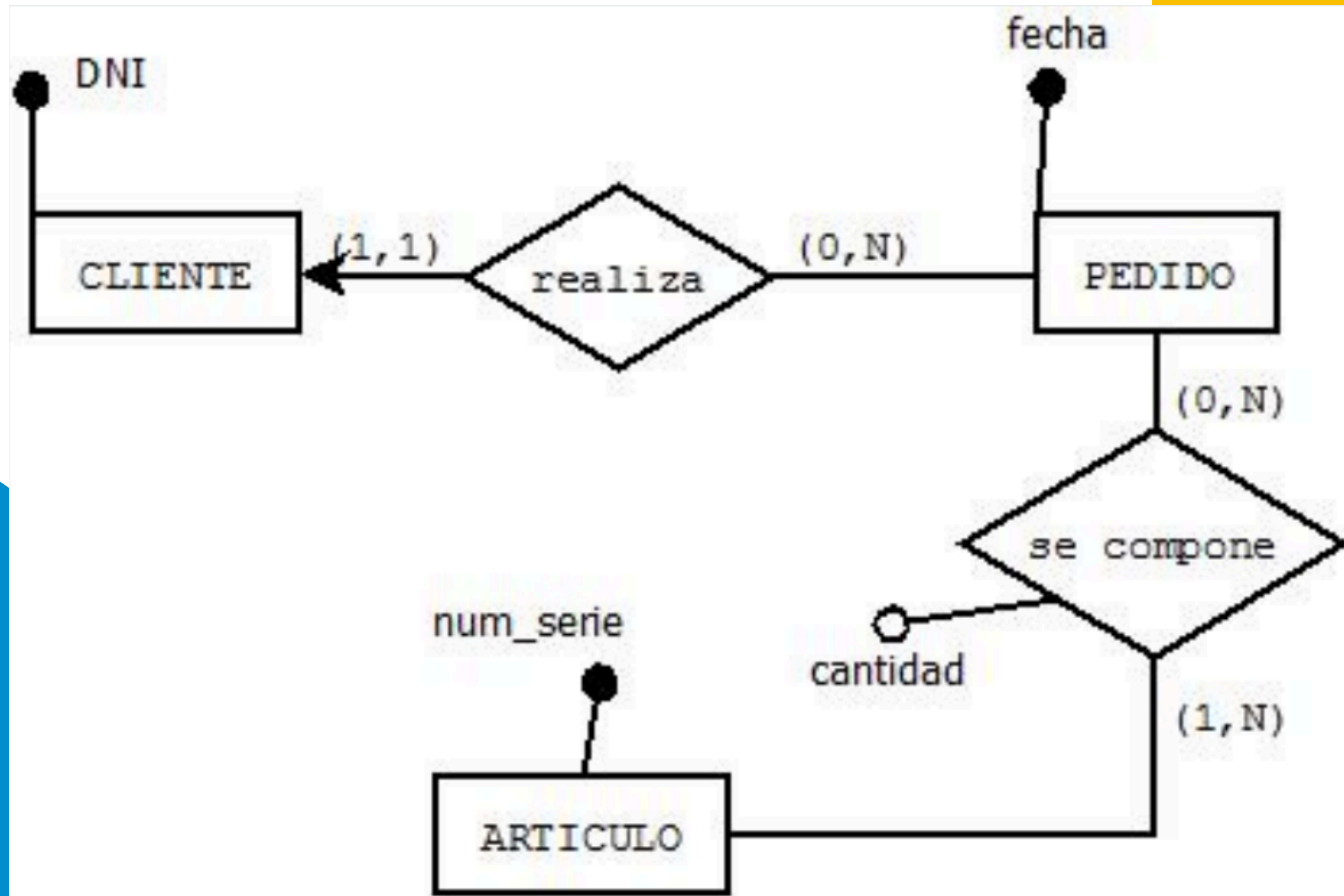
EJERCICIOS



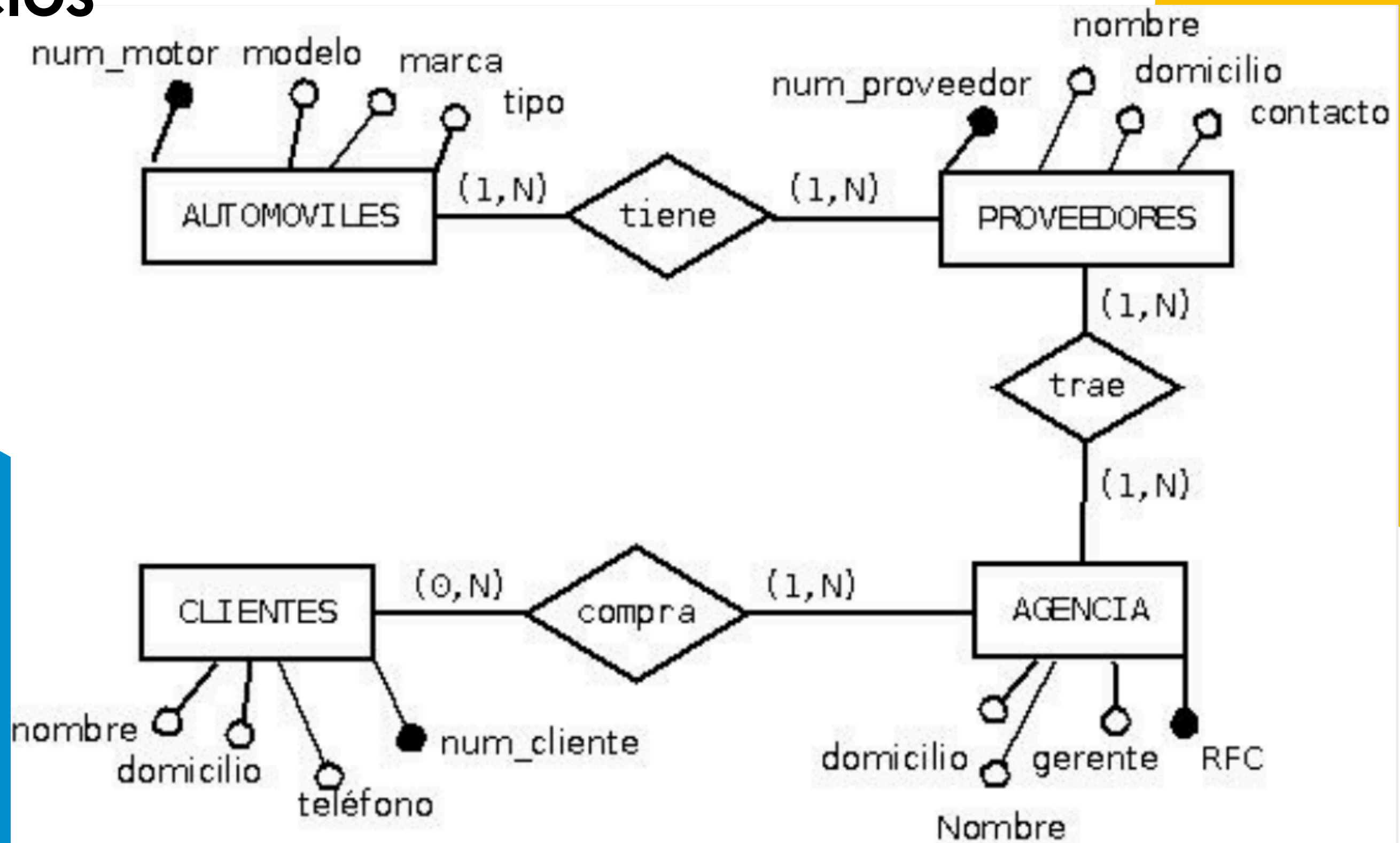
EJERCICIOS



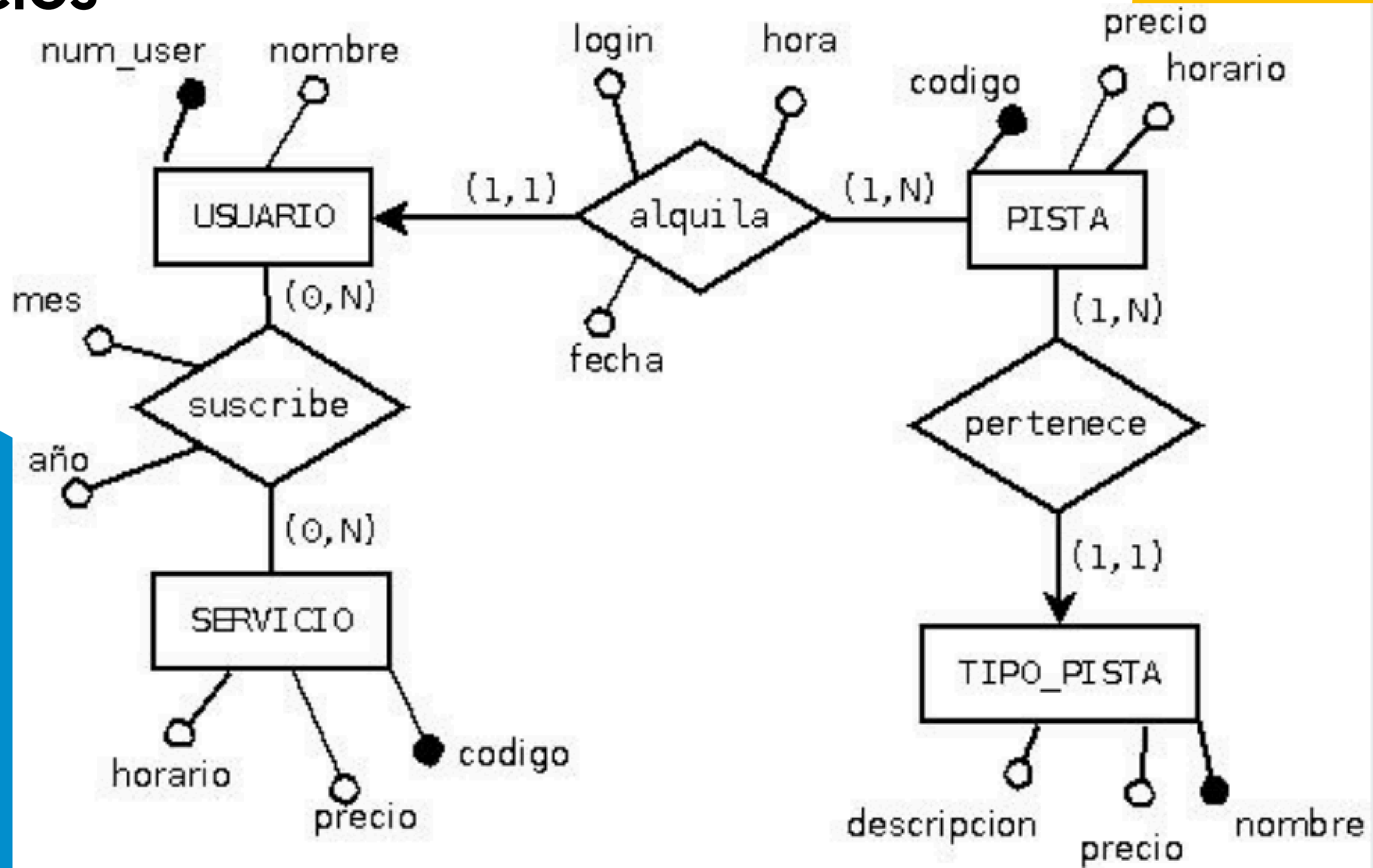
EJERCICIOS



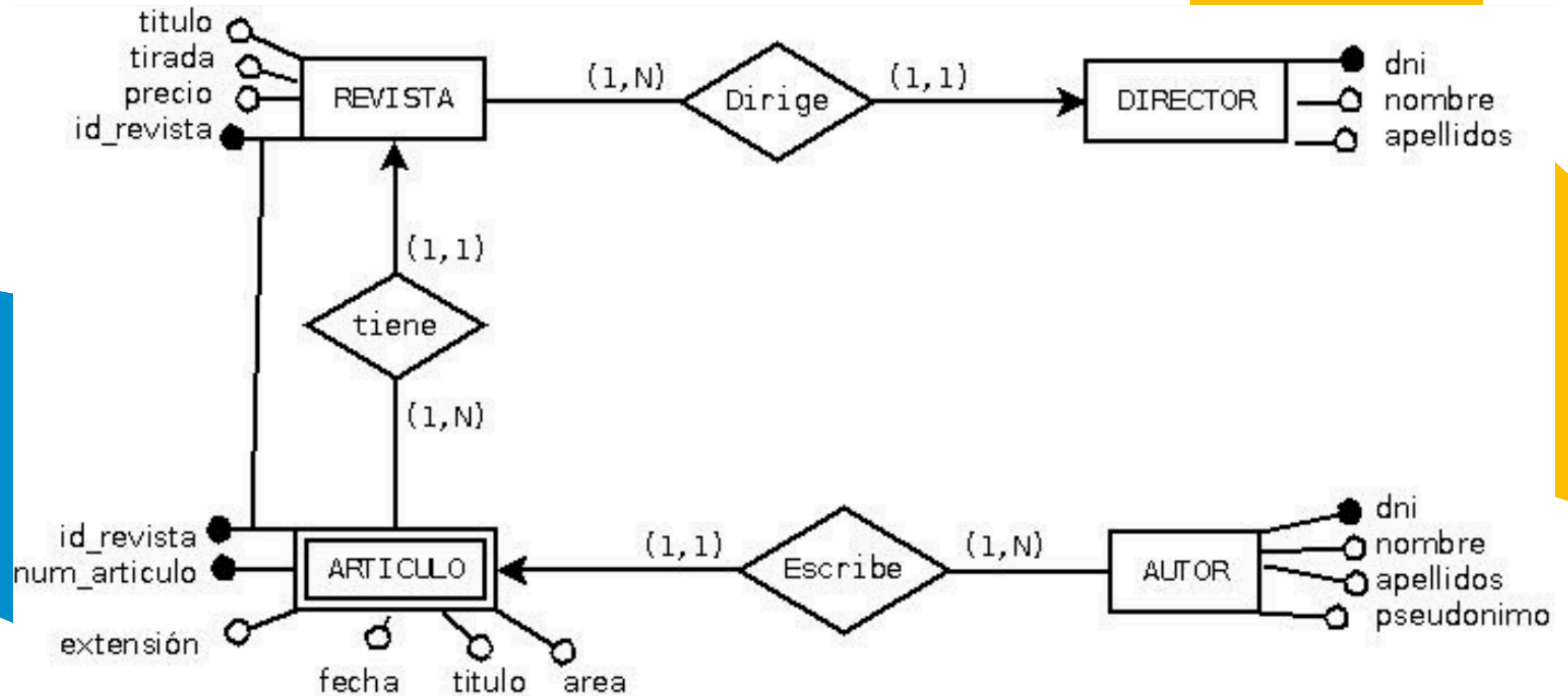
EJERCICIOS



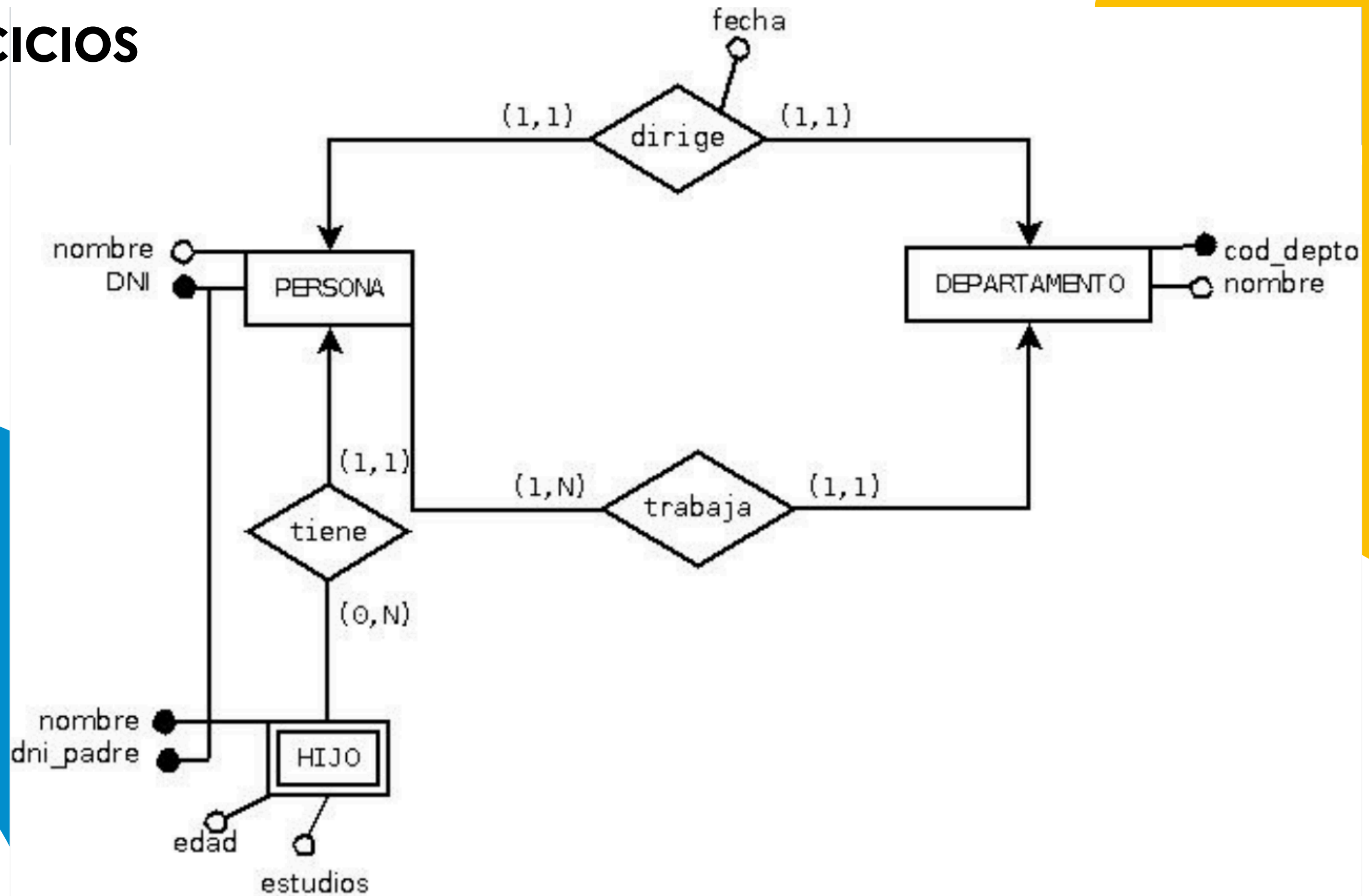
EJERCICIOS



EJERCICIOS



EJERCICIOS





RELACIONES REFLEXIVAS

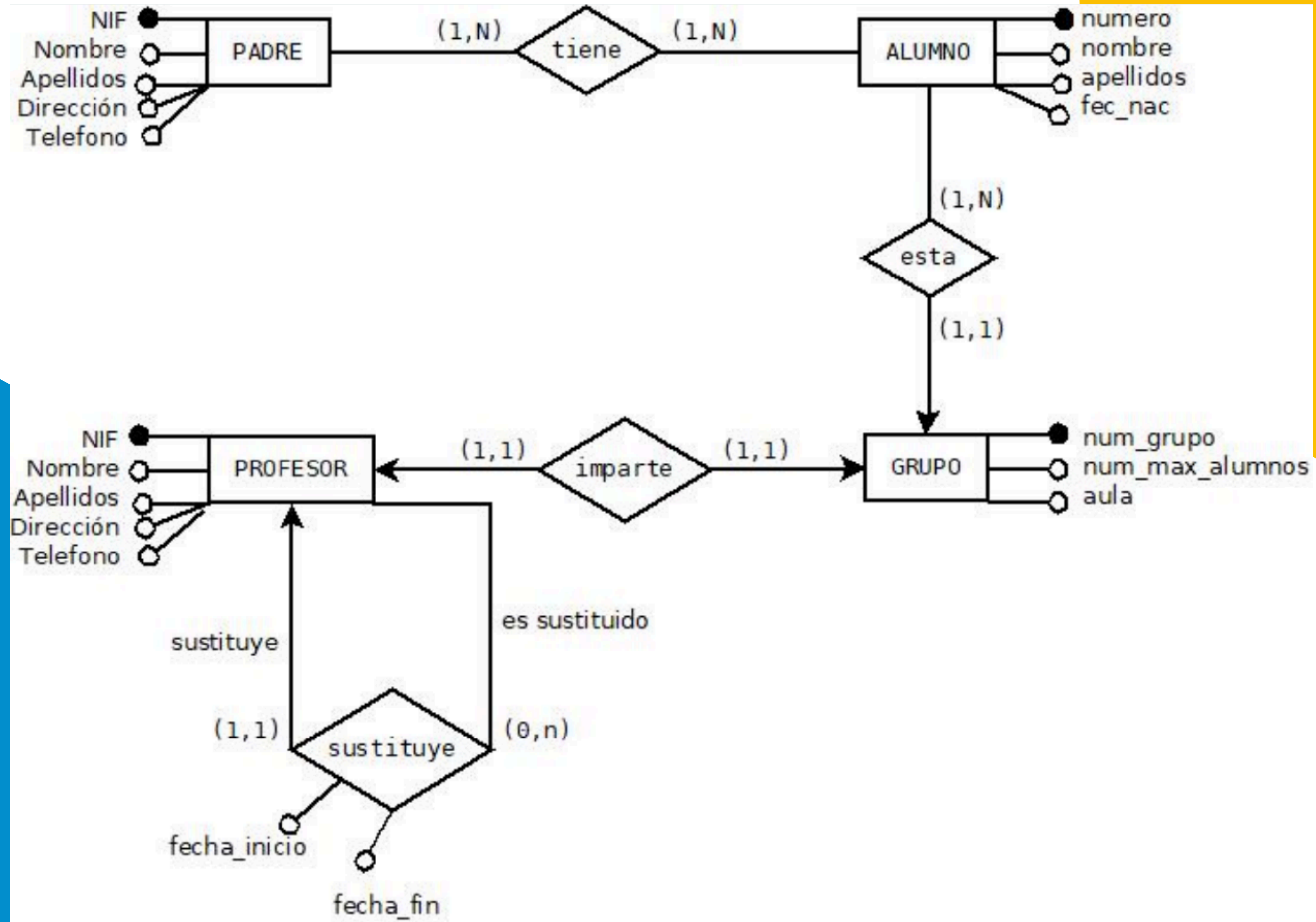
RELACIONES REFLEXIVAS

Son relaciones normales por lo que:

- **Nombre de la tabla:** nombre de la relación.
- **Columnas:** atributos propios de la relación.
- **Clave principal:** unión de las claves principales de las entidades que une
 - **Renombrar una de las claves.**

Se deberá estudiar la cardinalidad y (en caso necesario) la participación para ver si la relación se mantiene o desaparece.

EJERCICIOS



JERARQUÍAS

JERARQUÍAS

De nuevo nos encontramos con distintos tipos de jerarquías:

- Total disjunta
- Total solapada
- Parcial disjunta
- Parcial solapada

Aparecen ahora 3 posibles soluciones, siempre buscando la que **menos huecos deje**.

SOLUCIÓN 1

Crear **una tabla** por la entidad superior y otro por cada subconjunto

- **Conjunto superior:** Tabla como el resto
- **Subconjuntos:** Tabla independiente con sus atributos propios más la clave principal del conjunto superior

SOLUCIÓN 2

Crear una tabla por subconjunto

- **Conjunto superior:** desaparece
- **Subconjuntos:** Tabla independiente con sus atributos propios más la clave principal del conjunto superior y todos los atributos del conjunto superior.

SOLUCIÓN 3

Crear una tabla para el conjuntos superior

- **Conjunto superior:** englobará todos los atributos de todos los subconjuntos **más** un atributo que añadiremos para indicar el subconjunto al que pertenece -> **tipo**
- **Subconjuntos:** desaparecen

JERARQUÍAS

La solución depende de si la jerarquía es **total** o **parcial**.

Buscaremos siempre generar el **mínimo número de huecos vacíos**.

- Jerarquía total -> podría eliminarse la entidad superior.
- Jerarquía parcial -> no se puede eliminar la entidad superior.

NORMALIZACIÓN

NORMALIZACIÓN

Es un proceso que consiste en una aplicación de pasos o normas, para obtener datos agrupados en diferentes tablas de forma que sea la estructura más óptima.

Las formas normales se corresponde a una teoría de normalización iniciada por el propio Codd y continuada por otros autores.

Codd definió en 1970 la primera forma normal, desde ese momento aparecieron la segunda, tercera, la Boyce- Codd, la cuarta y la quinta forma normal.

PRIMERA FORMA NORMAL (1FN)

Tenemos dos definiciones:

“Una tabla está en 1FN si y sólo si los valores que componen los atributos de una tupla, son atómicos”

Los valores de un atributo son atómicos cuando no aparecen valores repetidos, y por tanto, son elementales y únicos.

“Una tabla se dice que está en 1FN si todos los atributos no clave, dependen funcionalmente de la clave”

DEPENDENCIA FUNCIONAL (DF)

Se dice que el atributo o conjunto de atributos B **depende funcionalmente** del atributo o conjuntos de atributos A, y se representa como **$A \rightarrow B$** , si y sólo si, **cada valor de A** se corresponde con un único **valor de B**.

COMO PASAR A 1FN

Para que una tabla que no está en 1FN se pase a 1FN se procede:

1. Se localizan los atributos que forman parte de la clave principal.
2. Se descompone la tabla con el uso de la Proyección:
 - a. Se deja la tabla original con la clave y los atributos que tienen valores únicos.
 - b. Se crea otra tabla con la clave y los atributos que tienen valores múltiples (Se distribuyen cada uno en una tupla distinta). Si la clave no está clara, se crea una.

DEPENDENCIA FUNCIONAL COMPLETA (DFC)

Se aplica a **claves primarias compuestas**. Se dice que el atributo Y de la relación R tiene una **dependencia completa** con el atributo X , si tiene una **dependencia funcional con X** y **no** depende funcionalmente de **ningún subconjunto de X** .

SEGUNDA FORMA NORMAL (2FN)

Una tabla está en 2FN si y sólo si se cumplen las condiciones:

1. Se encuentra en **1FN**.
2. Todo atributo **no primo** (que no pertenece a la clave primaria) **depende completamente** de la clave primaria.

Esta forma normal solo se considera si la Clave Primaria es **compuesta**, si no, ya está en 2FN.

COMO PASAR A 2FN

Descomponemos la tabla en 2:

1. Creamos una tabla con la Clave Primaria completa y todos los atributos no primos que dependan completamente.
2. Creamos otra tabla con la parte de la Clave Primaria que tiene dependencias, junto con los atributos no primos afectados.

TERCERA FORMA NORMAL (3FN)

Una relación está en 3FN si y sólo si se cumplen las condiciones:

1. Se encuentra en **2FN**.
2. No existen atributos no primos que sean **transitivamente dependientes** de cada posible clave de la tabla.

Esto quiere decir que un atributo **no primo**, sólo se puede conocer a través de la **clave principal** de la tabla y **no por medio** de otro atributo **no primo**.

DEPENDENCIA FUNCIONAL TRANSITIVA (DFT)

Sean tres subconjuntos distintos de atributos **A**, **B** y **C** pertenecientes a una tabla **T**, de tal modo que se cumplen las condiciones: **A** \rightarrow **B** y **B** \rightarrow **C**. Se dice que **C** tiene una **dependencia funcional transitiva** con **A**.

COMO PASAR A 3FN

Descomponemos la tabla en 2:

1. Creamos una tabla con la Clave Primaria completa y todos los atributos no primos que no producen conflicto, junto con el atributo no primo con el que se puede identificar a otros atributos no primos.
2. Creamos otra tabla con los atributos que producían conflicto, y el atributo no primo que se quedó en la tabla anterior, siendo éste la clave primaria.