



El modelo relacional (modelo lógico)

Curso 2022-2023



Introducción

Objetivo

- Proteger al usuario de la obligación de conocer las estructuras de datos físicas con las que se representa la información de una base de datos.
- Permite que la **BBDDs** se pueda implementar **en cualquier SGBD**.

Características fundamentales:

- La relación es el elemento fundamental del modelo.
 - BBDDs como colección de relaciones.
- El modelo relacional es independiente
 - De la forma en que se almacenan los datos
 - De la forma de representar los datos
- Se puede demostrar la eficacia del modelo a la hora de operar conjuntos de datos.

Las relaciones en el modelo relacional

- **Relación:**
 - Conjunto de atributos, cada uno de los cuales pertenece a un dominio, y que posee un nombre que identifica la relación.
 - Se representa por una tabla con columnas (atributos) y filas (tuplas).
 - Conjunto de **tuplas**:
 - Representa el cuerpo de la relación
 - Conjunto de **atributos** y el **nombre**:
 - Representan el esquema

Otros conceptos

Atributo:

- Características que describen a un entidad o relación

Dominio:

- Conjunto de valores permitidos para un atributo.
- Cadenas de caracteres, números enteros, booleanos, ...

Restricciones de semántica

- Condiciones que deben cumplir los datos para su correcto almacenamiento
 - **Restricciones de clave**
 - Conjunto de atributos que identifican de forma única a una entidad.
 - **Restricciones de valor único (UNIQUE)**
 - Impide que un atributo tenga un valor repetido (sin tener que ser clave)
 - **Restricciones de integridad referencial**
 - Una tabla tiene una referencia a algún valor de otra tabla.
 - Exige que exista el valor referenciado en la otra tabla
 - Ej.: No se puede poner una nota a un alumno que no exista.

Otros conceptos

- **Restricciones de semántica**
 - Condiciones que deben cumplir los datos para su correcto almacenamiento
 - **Restricciones de dominio**
 - Exige que el valor que puede tomar un campo esté dentro del dominio definido.
 - Ej.: DNI – 9 dígitos y una letra
 - **Restricciones de verificación (CHECK)**
 - Permite comprobar si un valor de un atributo es válido conforme a una expresión.
 - **Restricciones de valor NULO (NULL o NOT NULL)**
 - Atributo obligatorio si no admite el valor NULO o NULL
 - Si admite como valor el valor NULL, el atributo es opcional
 - **Disparadores o triggers:**
 - Procedimientos que se ejecutan para hacer una tarea concreta en el momento de:
 - Insertar, modificar o eliminar información de la tabla
 - **Restricciones genéricas adicionales o aserciones (ASSERT)**
 - Permite validar cualquiera de los atributos de una o varias tablas.

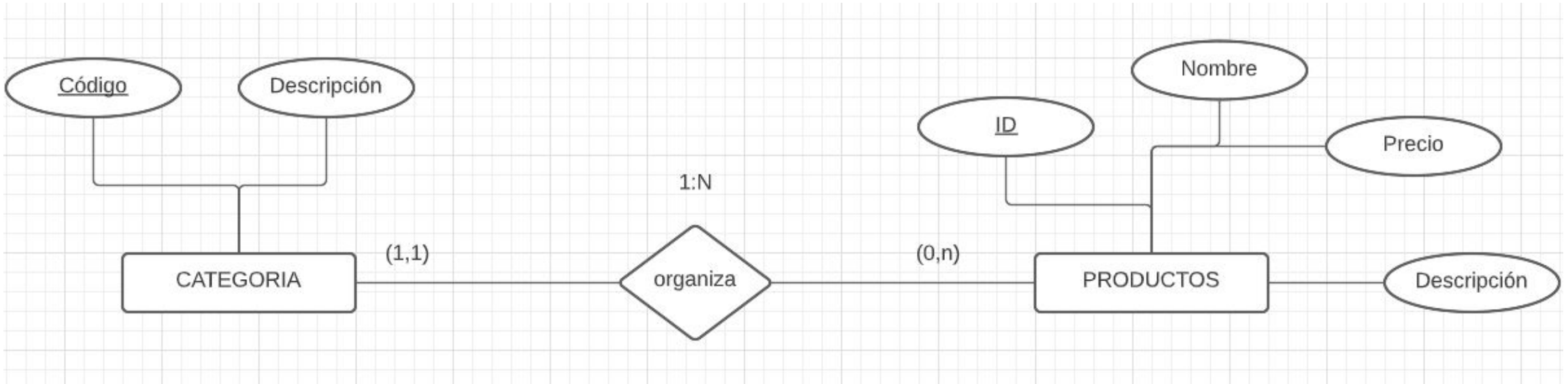
Otros conceptos

- **Clave**

- Conjunto de atributos que identifican de forma única una ocurrencia de entidad.
- Pueden ser simples (atómicas) o compuestas.
- Hay varios **tipos de clave**:
 - **Superclave**
 - Identifican a una entidad (pueden ser no mínimas)
 - Para empleado:
 - DNI, O el DNI+Nombre o el DNI+Nombre+Número de la Seguridad Social
 - **Clave Candidata**
 - Es la mínima superclave (DNI, Número de SS)
 - **Clave Primaria**
 - Clave candidata elegida por el diseñador como clave definitiva (DNI)
 - **Clave foránea**
 - Atributo de una entidad, que es clave en otra entidad.
 - Ejemplo: DNI es una clave foránea en la tabla notas.

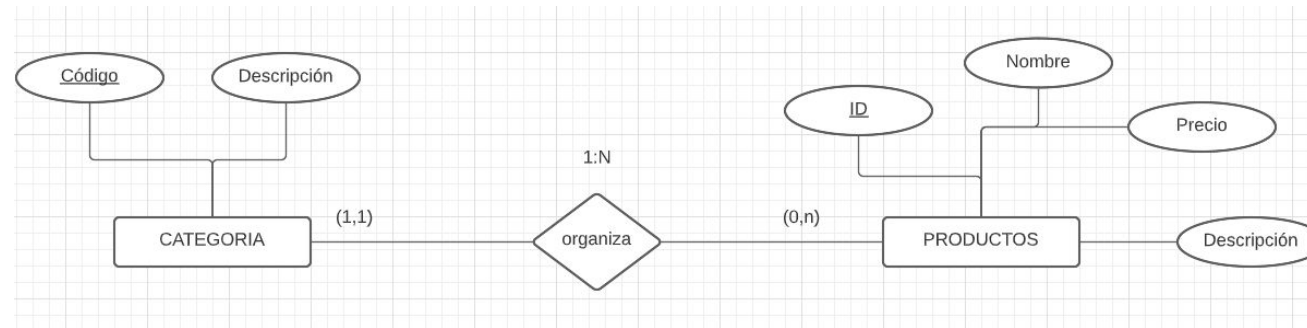
Transformación de un diagrama E/R al modelo relacional

- Transformación de entidades fuertes
 - Para cada entidad A, entidad fuerte, con atributos (a1, a2, ..., an)
 - Se crea una tabla A (con el nombre en plural) con n columnas correspondientes a los atributos de A, donde cada fila de la tabla A corresponde a una ocurrencia de la entidad A.



Transformación de un diagrama E/R al modelo relacional

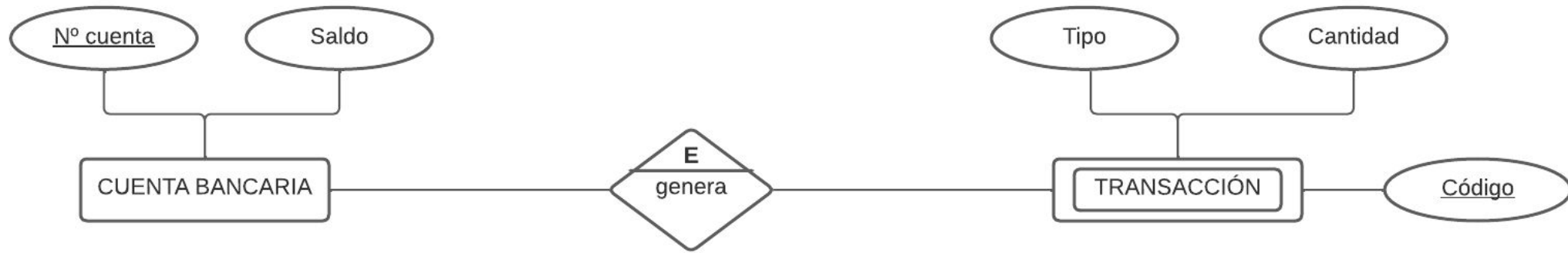
- Transformación de entidades fuertes:
 - Diagrama de E-R



- Las **tablas generadas** son:
 - CATEGORÍAS (Código, Descripción)
 - PRODUCTOS (Id, Nombre, Precio, Descripción)

Transformación de un diagrama E/R al modelo relacional

- Transformación de **entidades débiles**
 - Para cada entidad débil **D**, con atributos $cd_1, cd_2, \dots, cd_t, d_{t+1}, d_{t+2}, \dots, d_n$
 - Donde cd_1, cd_2, \dots, cd_t son los atributos clave de la entidad D
 - Una entidad fuerte **F** de la que depende D, con atributos clave (cf_1, cf_2, \dots, cf_m)
 - Se crea una tabla D con $m+n$ columnas $cd_1, cd_2, \dots, cd_t, d_{t+1}, d_{t+2}, \dots, d_n, cf_1, cf_2, \dots, cf_m$ correspondientes a los atributos de D y a los atributos clave de F.
 - Si la entidad débil, solo tiene dependencia de existencia
 - La clave primaria de la tabla D será la unión de los atributos clave de la entidad D.
 - Si además, tiene una dependencia de identificación
 - La clave primaria de la tabla D será la unión de los atributos $cd_1, cd_2, \dots, cd_t, cf_1, cf_2, \dots, cf_m$
 - La unión de los atributos clave de la entidad débil D y de la entidad fuerte F.



Transformación de un diagrama E/R al modelo relacional

• Las **tablas generadas** son:

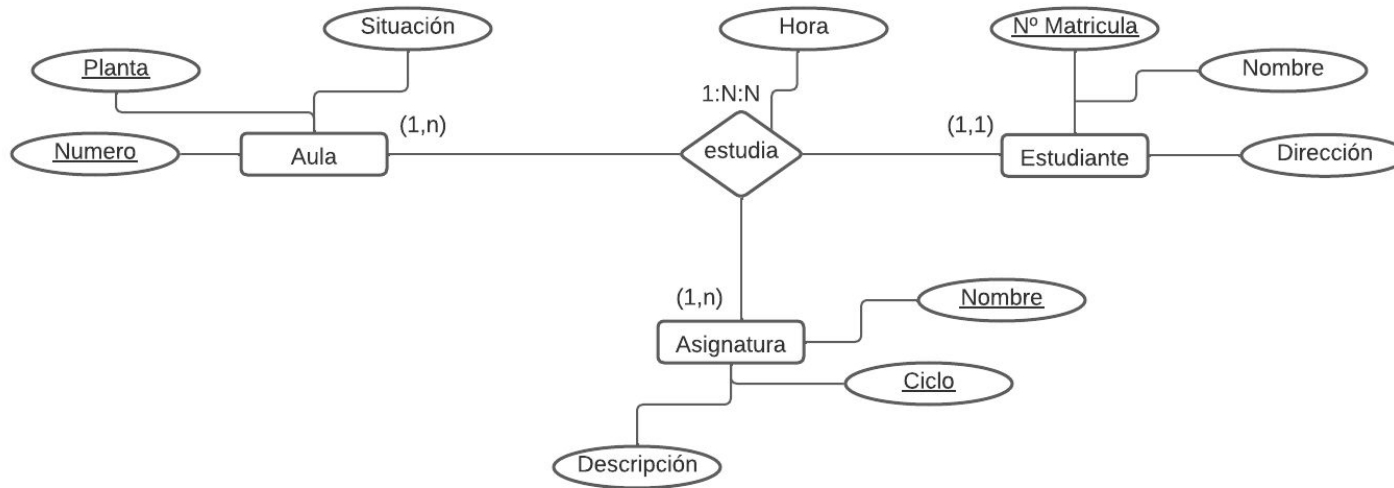
- CUENTAS_BANCARIAS (NºCuenta, saldo)
- TRANSACCIONES (Código, Tipo, Cantidad, Nº Cuenta)

Transformación de un diagrama E/R al modelo relacional

- Transformación de relaciones
 - Por cada relación R entre entidades E_1, E_2, \dots, E_N .
 - La clave de E_i es $C_i = a_{i1}, a_{i2}, \dots, a_{iN}$.
 - Regla general para las relaciones
 - Se crea una tabla con todos los campos clave de las entidades relacionadas y los atributos de la relación.
 - La clave primaria de la tabla generada es la suma de los atributos claves de las entidades relacionadas, y
 - Cada clave incorporada a la tabla, será una clave foránea que referencia a la tabla de la que se importa.

Transformación de un diagrama E/R al modelo relacional

- Transformación de relaciones



AULAS(Numero, Plantas, Situación)

ESTUDIANTES(Nº Matricula, Nombre, Direccion)

ASIGNATURAS(Nombre, Ciclo, Descripcion)

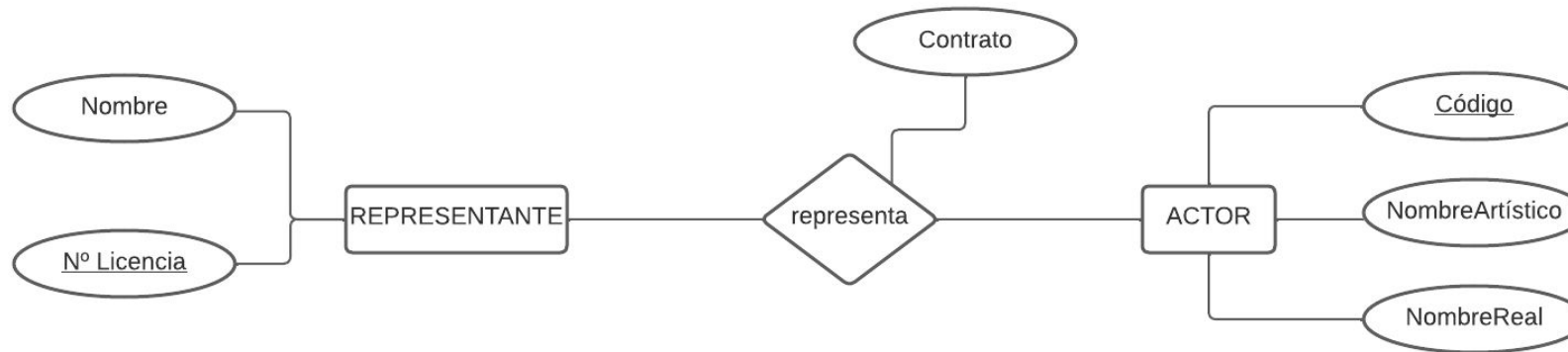
ESTUDIOS(Numero, Planta, NºMatricula, Nombre, Ciclo, Hora)

Transformación de un diagrama E/R al modelo relacional

- No siempre se aplica la regla general para crear una tabla por cada relación.
- Siguiendo **excepciones** a la regla general:
 - **Relaciones con cardinalidad 1:N**
 - Se añade a la tabla de la entidad que actúa con participación máxima N la clave de la entidad que actúa con participación máxima 1 (como clave foránea).
 - Si además, la relación tuviera atributos se importarían también a la entidad que actúa con participación máxima N

Transformación de un diagrama E/R al modelo relacional

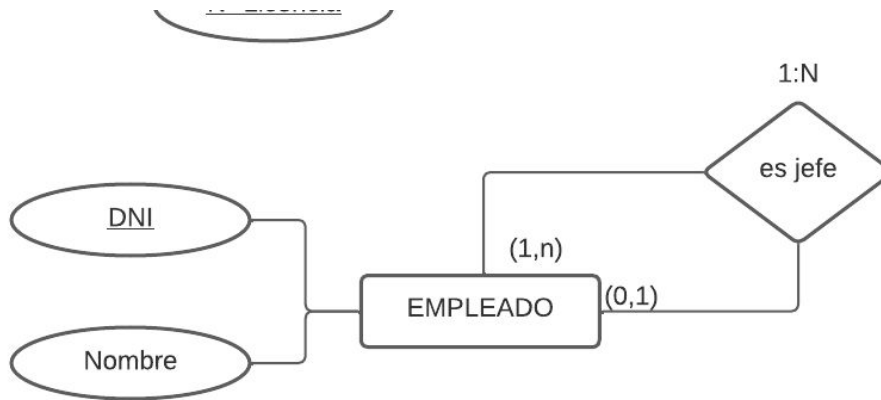
- **Excepciones**
 - Relaciones con cardinalidad 1:N



ACTORES(Código, NombreArtístico, NombreReal, N_Licencia_Representante, Contrato)
REPRESENTANTES(N_Licencia, Nombre)

Transformación de un diagrama E/R al modelo relacional

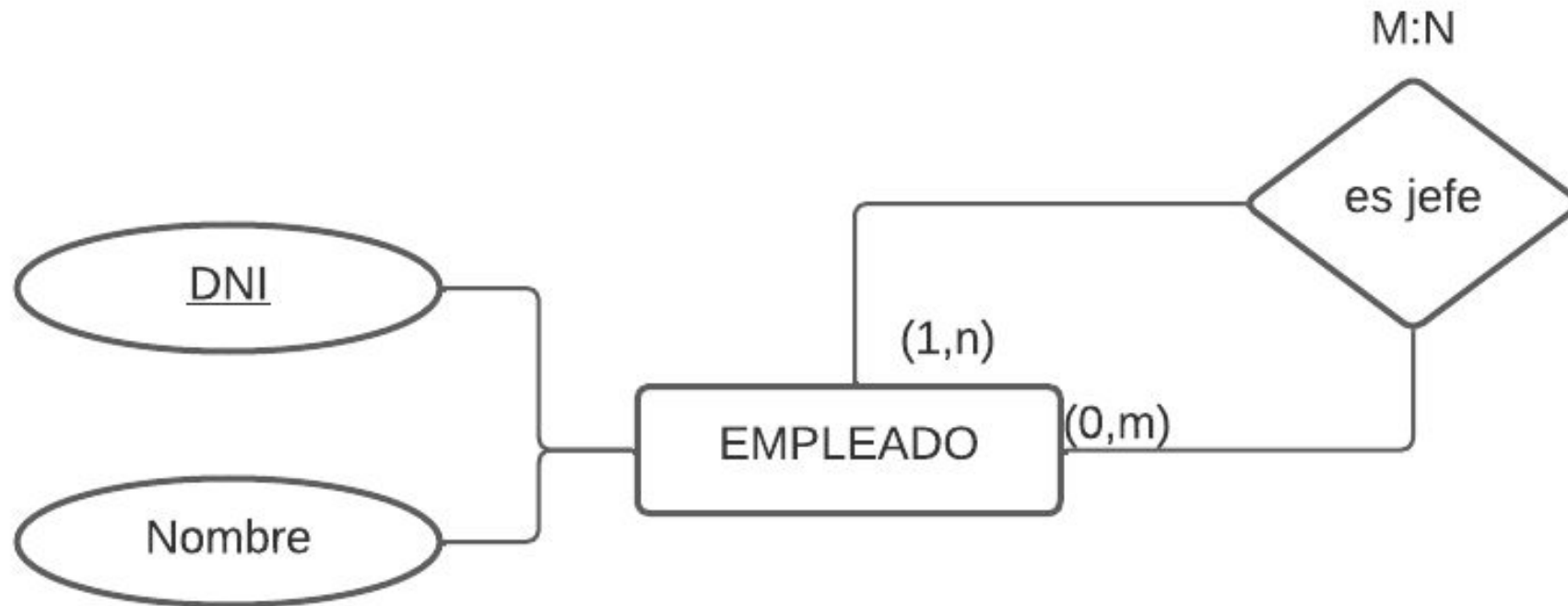
- Siguiendo **excepciones** a la regla general:
 - **Relaciones reflexivas con cardinalidad 1-N**
 - Hay que crear una tabla con el nombre de la entidad, añadiendo otra vez la clave cambiada de nombre.



EMPLEADOS(DNI, Nombre, DNISupervisor)

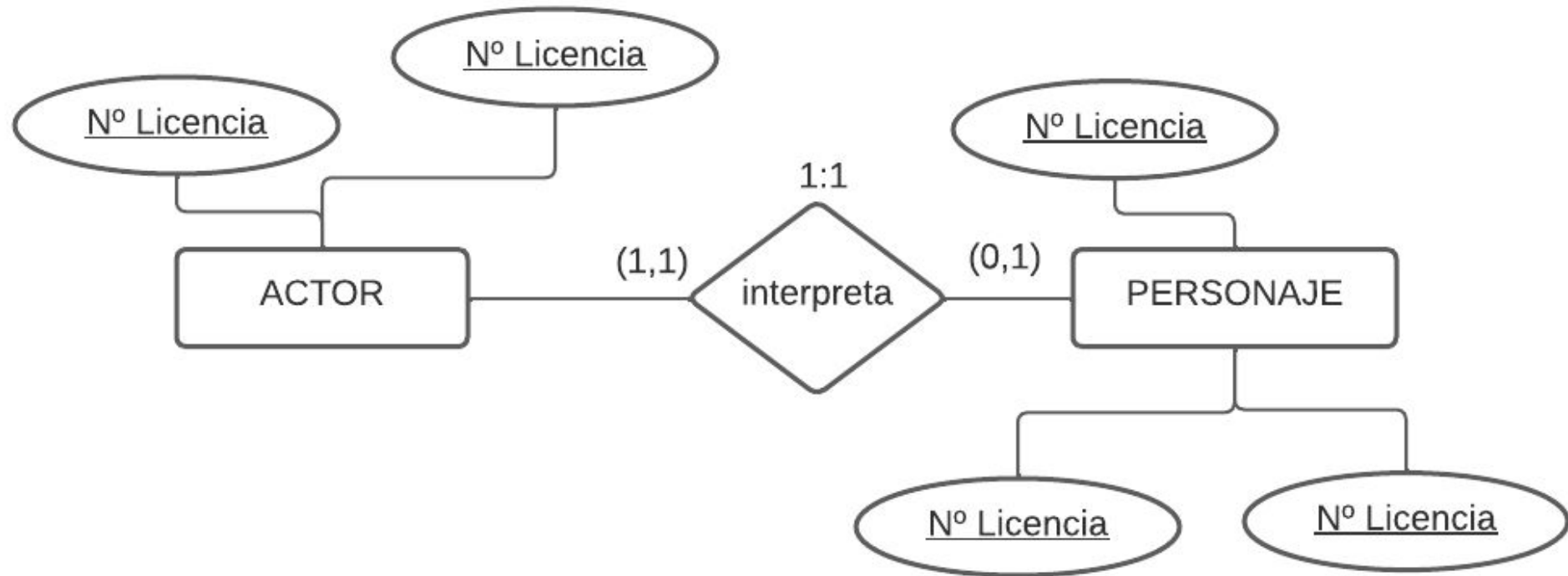
Actividad

- En las relaciones reflexivas con cardinalidad m-n, se aplica la regla general para la transformación de relaciones. Expresa cómo sería la regla para crear tablas con relaciones reflexivas con cardinalidad m-n. Después, aplica esa regla para transformar la siguiente relación, suponiendo que tuviera cardinalidad m-n.



Transformación de un diagrama E/R al modelo relacional

- No siempre se aplica la regla general para crear una tabla por cada relación.
- Sigüientes **excepciones** a la regla general:
 - **Relaciones 1-1**
 - Se tiene la libertad de poder incorporar la clave de una de las dos entidades a la otra



Transformación de un diagrama E/R al modelo relacional

- **Opciones**

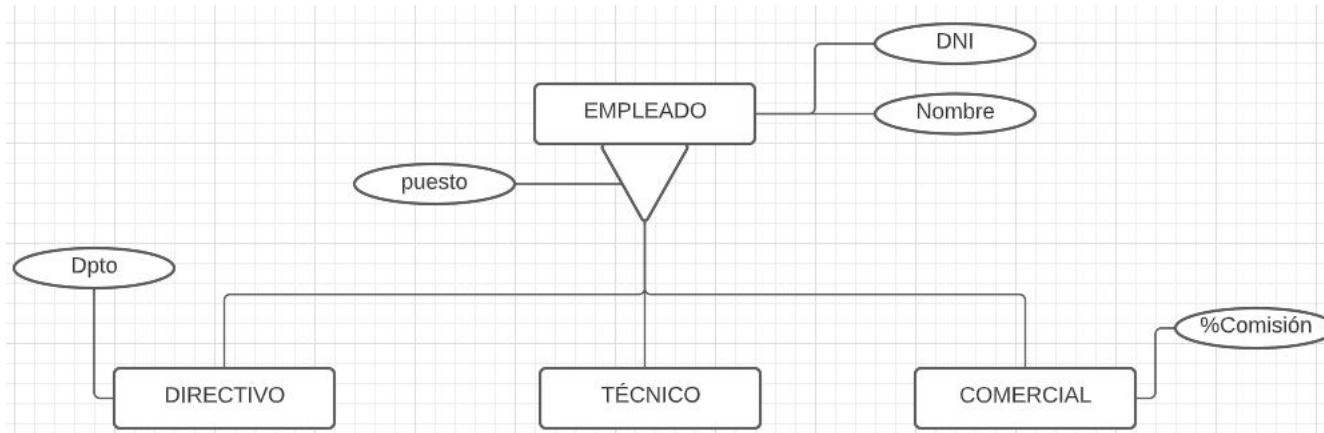
- Incorporar la clave de Personajes como clave foránea en la tabla actores:
 - ACTORES(Codigo, Nombre, CodigoPersonaje)
 - PERSONAJES(Codigo, Nombre, Película)
- Incorporar la clave de Actores como clave foránea en la tabla Personajes:
 - ACTORES(Codigo, Nombre)
 - PERSONAJES(Codigo, Nombre, Película, CodigoActor)
- Incorporar la clave de Actores como clave foránea en la tabla Personajes y la clave de Personajes a la tabla de Actores como clave foránea:
 - ACTORES(Codigo, Nombre, CódigoPersonaje)
 - PERSONAJES(Codigo, Nombre, Película, CodigoActor)

Transformación de un diagrama E/R al modelo relacional

- No siempre se aplica la regla general para crear una tabla por cada relación.
- Sigüientes **excepciones** a la regla general:
 - **Participaciones 0,x**
 - Muchas veces las participaciones se omiten y sólo se toma en cuenta la cardinalidad
 - Es necesario tener en cuenta cuando la participación tiene un mínimo de 0
 - Adoptar un campo de una tabla como opcional NULL, u obligatorio NOT NULL

Transformación de un diagrama E/R al modelo relacional

- No siempre se aplica la regla general para crear una tabla por cada relación.
- Sigüientes **excepciones** a la regla general:
 - **Generalizaciones y especializaciones**
 - Se puede optar por 4 opciones. Cada opción se adaptará mejor o peor a los diferentes tipo de especialización (Exclusiva, Inclusiva, Total, Parcial)



Transformación de un diagrama E/R al modelo relacional

1. Una tabla para la superclase y otras tantas para cada subclase, incorporando el campo clave de la superclase a las tablas de la subclases.
 - EMPLEADOS (DNI, Nombre, Puesto)
 - DIRECTIVOS(DNI, Dpto)
 - TECNICOS(DNI, Máquinas)
 - COMERCIALES(DNI, Comisión)
2. Crear una tabla para cada subclase incorporando todos los atributos de la clase padre, y no crear una tabla para la superclase:
 - DIRECTIVOS(DNI, Nombre, Puesto, Dpto)
 - TECNICOS(DNI, Nombre, Puesto, Máquinas)
 - COMERCIALES(DNI, Nombre, Puesto, Comisión)

Transformación de un diagrama E/R al modelo relacional

1. Crear una sola tabla para la superclase, incorporando los atributos de todas las subclases y añadir, para distinguir el tipo de la superclase, un campo llamado “tipo”, que contendrá el tipo de la subclase al que representa cada tupla. (**Especializaciones exclusivas**)
 - EMPLEADOS(DNI, Nombre, Puesto, Dpto, Máquinas, Comisión, Tipo)
2. Crear una sola tabla con la opción anterior, pero en lugar de añadir un solo campo “tipo”, se añaden varios campos que indiquen si cumple un perfil, de este modo se soportan las **especializaciones inclusivas**.
 - EMPLEADOS(DNI, Nombre, Puesto, Dpto, Máquinas, Comisión, EsDirectivo, EsTécnico, EsComercial)

Normalización

- Después de realizar la transformación del modelo conceptual (E/R) al lógico(relacional).
- Medir **calidad** de lo construido:
 - Uno de los parámetros que mide la calidad de una BBDDs es la **forma normal**
 - Esta **forma normal** puede alcanzarse cumpliendo ciertas **restricciones** que impone **cada forma normal** al **conjunto de atributos** de un diseño.
 - El **proceso** de **obligar** a los **atributos** de un diseño a **cumplir ciertas formas normales** se llama **normalización**.

Normalización

Objetivos que pretenden alcanzar las formas normales:

- **Evitar la redundancia de datos.** (Guardar cada hecho sólo una vez)
 - Reduciendo el espacio de almacenamiento
- Que los hechos distintos se almacenen en sitios distintos.
 - Evitando ciertas anomalías a la hora de operar con los datos.

Hay 6 formas normales

- Cada forma agrupa a las anteriores
 - La forma normal 3 cumple la forma normal 2 y la forma normal 1.

Conceptos para normalización

- **Dependencia funcional**

- Un atributo Y depende funcionalmente de otro atributo X, o que $X \twoheadrightarrow Y$, si cada valor de X tiene asociado en todo momento un único valor de Y.
- También se dice que X implica Y, y por tanto, que X es el implicante.
- Ejemplo:
 - PRODUCTOS(CódigoProducto, Nombre, Precio, Descripción)
 - CódigoProducto \twoheadrightarrow Nombre,
 - Un código de producto sólo puede tener asociado un único nombre.
 - A través del código de producto se localiza un único nombre.

Conceptos para normalización

- **Dependencia funcional completa**

- Dado una combinación de atributos $X(X_1, X_2, \dots)$, se dice que Y tiene dependencia funcional completa de X , o que $X \twoheadrightarrow Y$, si depende funcionalmente de X , pero no depende de ningún subconjunto del mismo.
- Ejemplo:
 - COMPRAS(CódigoProducto, CódigoProveedor, Cantidad, FechaCompra)
 - CódigoProducto, CódigoProveedor \twoheadrightarrow FechaCompra

Conceptos para normalización

- **Dependencia transitiva**

- Dada la tabla T, con atributos (X, Y, Z) donde $X \rightarrow Y$, $Y \rightarrow Z$ e $Y \not\rightarrow X$, se dice que X depende transitivamente de Z, o que, $X \twoheadrightarrow Z$.
- Ejemplo:
 - PRODUCTOS(CódigoProducto, Nombre, Fabricante, País)
 - CódigoProducto \rightarrow Fabricante
 - Fabricante \rightarrow País
 - CódigoProducto \twoheadrightarrow País, es decir, CódigoProducto depende transitivamente de País.
 - PRODUCTOS(CódigoProducto, Nombre, Fabricante, País)
 - CódigoProducto \rightarrow Nombre
 - Nombre \rightarrow CódigoProducto
 - CódigoProducto $\not\rightarrow$ Nombre

Formas normales

- FN 1:
 - Se prohíbe que en una tabla haya **atributos** que puedan tomar **más de un valor**.
- FN 2:
 - Si está en FN 1 y además, cada atributo que no forma parte de la clave tiene dependencia completa de la clave principal.
- FN3:
 - Si está en FN2 y además, no hay ningún atributo no clave que dependa de forma transitiva de la clave.
- FNBC (Boyce-Codd)
 - El modelo está en FN3, y que además, todo implicante de la tabla, sea una clave candidata.

Actividad

- Entender bien las formas normales y poner un ejemplo de cada una.
 - Tendréis que ser capaces de explicar (1FN, 2FN, 3FN y FNBC)