

Tema 4 – Transformación del Modelo E/R al Modelo Relacional

1º DAW – Bases de Datos

J. Carlos Moreno

Curso 2023/2024

Tabla de contenido

4	Transformación del Modelo E/R al Modelo Relacional	2
4.1	Introducción.....	2
4.2	Transformación de Entidades	3
4.2.1	Transformación de los Atributos de las Entidades	3
4.2.2	Transformación de Atributos Compuestos	4
4.2.3	Transformación de los Atributos Multivaluados	5
4.2.4	Caso Práctico I. “Transformación Entidad Cliente”	6
4.3	Transformación de las Relaciones M:N	7
4.4	Transformación de las Relaciones 1:N	8
4.4.1	Caso A. (1,1).....	10
4.4.2	Caso B. (0,1).....	10
4.5	Transformación de las Relaciones 1:1	11
4.5.1	Caso A. E1 (1,1) y E2 (1,1).....	11
4.5.2	Caso B. E1(0,1) y E2(0,1)	12
4.5.3	Caso C. E1 (0,1) y E2 (1,1).....	13
4.6	Transformación de las Relaciones Ternarias	13
4.6.1	Caso A. M:N:P	14
4.6.2	Caso B. M:N:1	15
4.6.3	Caso C. M:1:1.....	16
4.6.4	Caso D. 1:1:1.....	17
4.7	Transformación de las Relaciones n-arias	18
4.8	Transformación de las relaciones Reflexivas	18
4.8.1	Caso A. 1:1 con (0,1) y (0,1).....	19
4.8.2	Caso B. 1:N o 1:1	19
4.8.3	Caso C. M:N	20
4.9	Transformación de las Relaciones Débiles	21
4.9.1	Dependencia en Existencia	21
4.9.2	Dependencia en Identificación	22
4.10	Transformación de las Jerarquías.....	22

4 Transformación del Modelo E/R al Modelo Relacional

4.1 Introducción

En el tema anterior estudiamos un modelo conceptual de datos que nos permitía describir la información que se desea almacenar en una base de datos cualquiera mediante el modelo Entidad-Relación. La ventaja de este modelo es que es independiente del modelo lógico sobre el que se vaya a implantar finalmente dicha base de datos. Por otro lado, cuando dicho modelo lógico es el modelo relacional, resulta bastante sencillo pasar del diagrama E-R al esquema relacional mediante unas reglas sencillas y fáciles de aplicar, llamadas Reglas de Transformación del Modelo E/R al Modelo Relacional

Antes de comenzar es necesario resaltar las diferencias existentes entre estos dos modelos.

- De una parte, el modelo E-R trabaja a nivel conceptual, estableciendo cuáles son las entidades fuertes y débiles que intervienen en nuestra base de datos, y las relaciones existentes entre ellas; sin embargo, no hace referencia alguna a la forma en que estos «objetos» se almacenan en ninguna base de datos, entre otras cosas por se trata sólo de un modelo conceptual.
- Por contra, el modelo relacional lo que trata es de representar la información en la forma en que se va a almacenar los sistemas de almacenamiento del ordenador (o al menos en la forma en que el usuario las percibe). Para ello se vale casi únicamente del concepto de tabla. Por tanto, lo que se pretende con este apartado es pasar de describir conceptualmente el mundo mediante entidades y relaciones, a describirlo lógicamente mediante tablas.

El diseño lógico se puede dividir en dos etapas:

- Diseño lógico estándar. Partiendo del esquema conceptual (en nuestro caso el modelo E-R), se elabora un esquema lógico estándar basado en el modelo de datos que soporte el SGBD elegido (generalmente, el modelo relacional).
- Diseño lógico específico. Una vez obtenido el diseño lógico estándar, se transforma al lenguaje de definición de datos soportado por el SGBD elegido (INGRES, ORACLE, DB2, ACCESS, MySQL, SYBASE, INFORMIX, etc.)



Ilustración 4-1. Esquema

4.2 Transformación de Entidades

Cada tipo de entidad se convierte en una Relación o Tabla en el modelo relacional. La relación se llamará igual que el tipo de entidad de donde proviene.

4.2.1 Transformación de los Atributos de las Entidades

Por regla general cada atributo de una entidad se transforma en un atributo (columna) de la tabla a la que ha dado lugar la entidad, aunque esta transformación dependerá del tipo de atributo:

- Identificador Principal o Atributos Clave. El (o los) atributo(s) clave(s) de una entidad pasan a ser la clave primaria de la tabla creada. Se representará en el Modelo Relacional en negrita y subrayado. A estos tipos de atributos se le aplicará la restricción PRIMARY KEY
- Atributos Obligatorios, no tienen una representación especial en el Modelo Relacional, simplemente su nombre. A estos tipos de atributos se les aplicará la restricción NOT NULL
- Atributos Opcionales, se muestra el nombre del atributo seguido de un asterisco
- Claves Alternativas, estos tipos de atributos se representan en el Modelo Relacional mediante un subrayado doble y se les aplicará la restricción UNIQUE.
- Atributos Dervidos, no tienen una representación especial en el Modelo Relacional, eso sí, se deberá realizar un procedimiento, función o Triggers para que calcule su valor de forma automática
- Mención especial merece la transformación de los atributos multivaluados

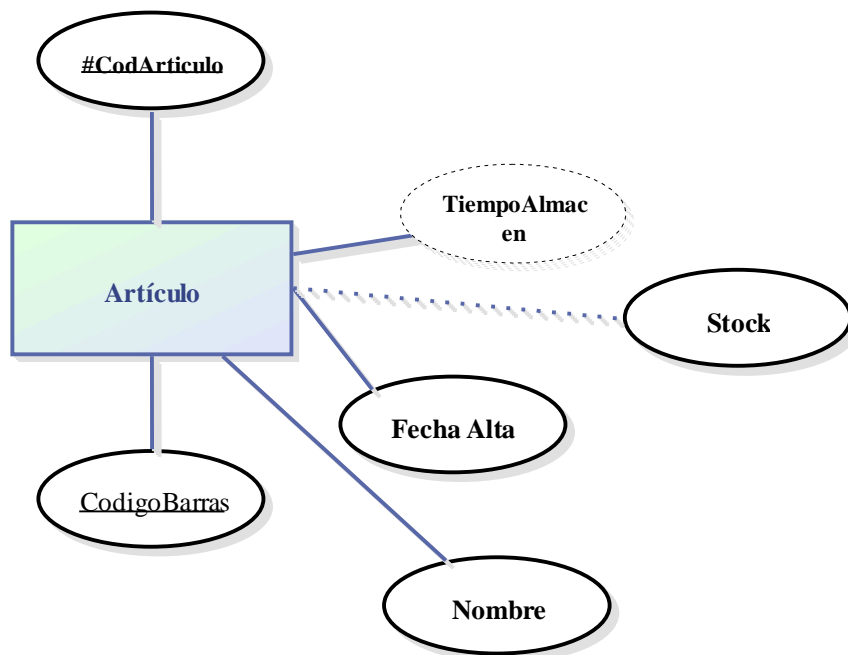


Ilustración 4-2. Transformación de Atributos

A partir de la figura 4-2 la transformación de la Entidad Artículo junto con los atributos quedaría de la siguiente forma:

Articulo(**#CodArtículo**, Nombre, CodigoBarras, FechaAlta, Stock*, TiempoAlmacen-d)

Cuando se implemente en SQL dicha tabla tendremos que aplicar las siguientes restricciones:

- PRIMARY KEY a CodArtículo
- UNIQUE a CódigoBarras
- NOT NULL FechaAlta, CódigoBarras, Nombre y TiempoAlmacen
- TRIGGERS a TiempoAlmacen

4.2.2 Transformación de Atributos Compuestos

Un atributo compuesto es aquél que puede dividirse en bloques de información más pequeños. El caso más común es el atributo Domicilio de algunas entidades. Normalmente en el Domicilio de un Cliente o de un Proveedor, no es necesario distinguir entre calle, número, portal, planta, etc. Si en nuestro diseño deseamos distinguir efectivamente entre todos esos componentes del Domicilio entonces la única solución es crear atributos atómicos para cada uno de ellos. Esta situación suele presentarse en raras ocasiones, y normalmente el diseño suele hacerse directamente en base a los atributos atómicos que pretenden distinguirse.

La Regla que aplicaremos será:

- El atributo compuesto no se incluye en la tabla o relación, solamente los atributos que lo componen.

Veamos el siguiente de la figura 4-3

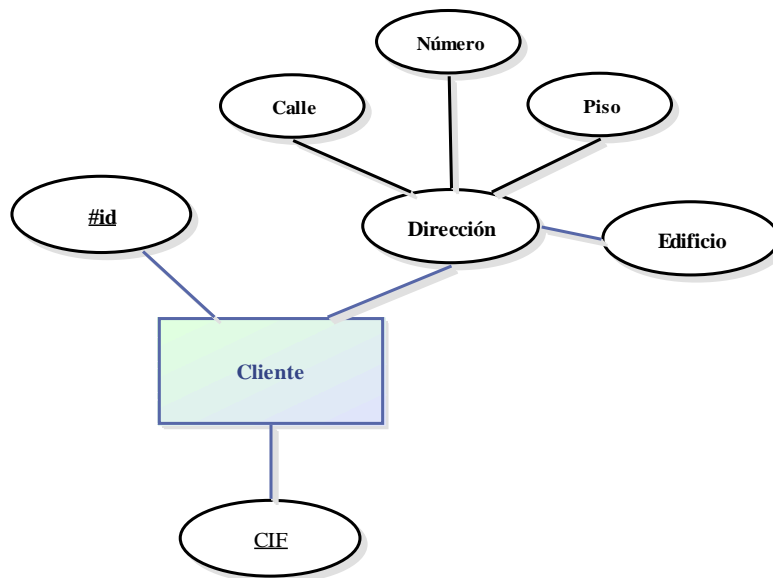


Ilustración 4-3

La transformación al Modelo Relacional sería:

- Cliente (#id, CIF, Calle, Número, Piso, Edificio)

4.2.3 Transformación de los Atributos Multivaluados

El Modelo Relacional impone que todos los atributos han de ser atómicos, es decir, sólo permiten almacenar un solo valor. Sin embargo en el Modelo Entidad Relación podemos representar atributos multivaluados, atributos que permitirían más de un valor, por ejemplo, teléfonos en la entidad Clientes, en el caso de que un Cliente tenga varios teléfonos.

Para poder realizar la transformación correcta al Modelo Relacional, previamente se deberá tener en cuenta la siguiente regla:

- Se crea una entidad débil con el nombre del atributo multivaluado y se relaciona mediante una dependencia en Identificación con la entidad principal, que pasará a ser la entidad fuerte de la relación.

Veamos el siguiente ejemplo

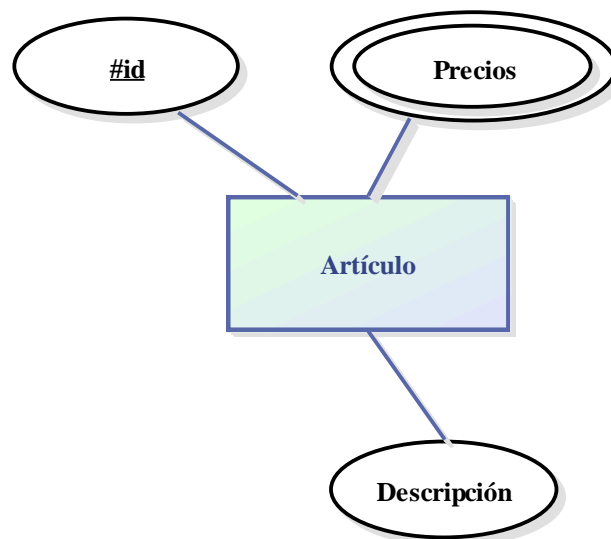


Ilustración 4-4. Atributos Multivaluados

Dado el Modelo anterior con el atributo multivaluado Precios, habría que crear una entidad débil con Precios, relacionada con la entidad fuerte Artículos mediante una dependencia en identificación. El nuevo modelo quedaría tal y como se muestra en la figura 4-4.

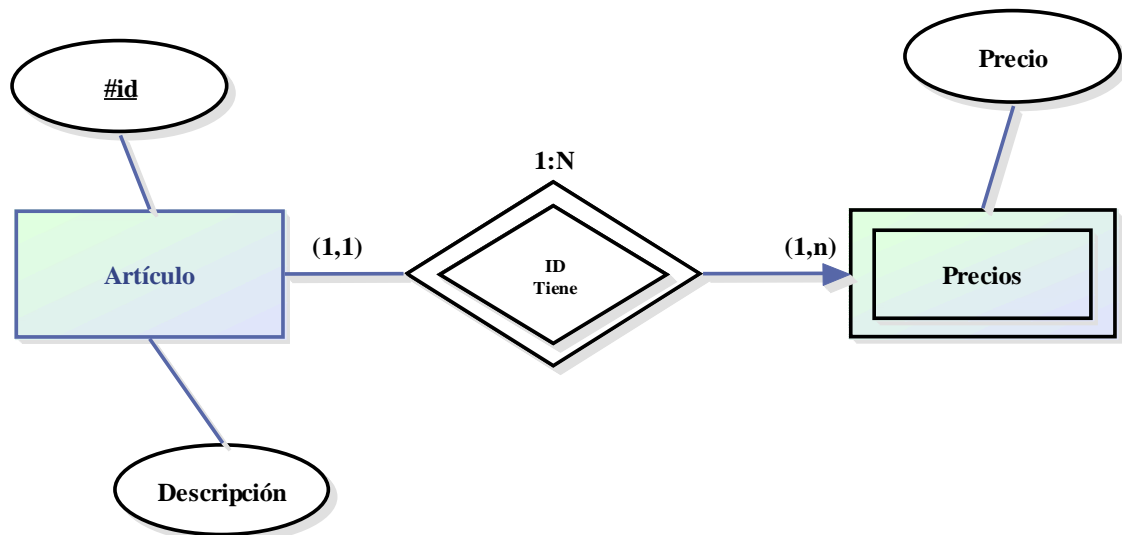


Ilustración 4-5. Transformación del atributo multivaluado Precios

Para la transformación al modelo relacional creamos las Tablas Artículos y Precios, sólo que la clave principal de Precios estará formada por la concatenación de CodArtículo + Precio, además CodArtículo sería clave ajena que referencia a la tabla Artículo con tipo de modificación en cascada y tipo de barrado también en cascada. Por lo tanto el Modelo Relacional quedaría de la siguiente forma:

Artículo (#id, Descripción)

Precios (#id_articulo, #Precio)

Id_articulo -> Artículo(MC:BC)

Cuando realicemos la implementación en SQL habrá que aplicar las siguientes restricciones:

- PRIMARY KEY a CodArtículo de Artículo
- PRIMARY KEY a (CodArtículo, Precio) de Precios
- NOT NULL Descripción
- FOREIGN KEY de Codartículo de Precios que referencia a la tabla Artículo

4.2.4 Caso Práctico I. “Transformación Entidad Cliente”

En un diagrama entidad relación nos encontramos con la siguiente entidad de clientes

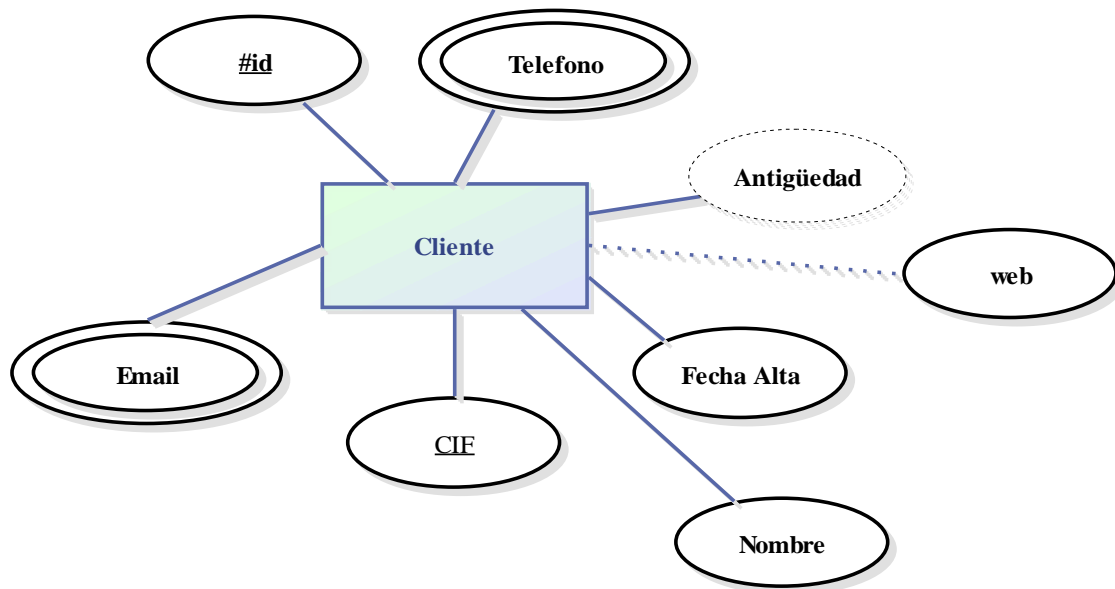


Ilustración 4-6. Entidad Cliente junto con sus atributos

Aplicando la regla de transformación de Entidades y Atributos, obtenemos:

- Cliente(**#id**, CIF, Nombre, Web*, Antigüedad-d, FechaAlta)
- TelefonosClientes(**#id_cliente**, **#Telefono**)
Id_cliente -> Cliente(MC, BC)
- EmailCliente(**#id_cliente**, **#Email**)
Id_cliente->Cliente(MC, BC)

4.3 Transformación de las Relaciones M:N

Es una de las reglas más sencillas de aplicar en el proceso de transformación del Modelo E/R al Modelo Relacional, puesto que no hay que tener en cuenta ningún caso excepcional, este tipo de relación viene marcado además porque implica la creación de una nueva tabla en el modelo.

La regla dice:

- Una relación M:N se transforma en una **nueva tabla** cuya clave primaria es la concatenación de las claves primarias de las entidades que forman parte de la relación.
- En la nueva tabla hay que incluir los atributos propios de la relación M:N si los tuviese
- Cada uno de los atributos que forman la clave primaria son a su vez claves ajenas (FOREIGN KEY) que referenciarán a las respectivas tablas de donde proceden.

Veamos el ejemplo de la figura 4-7.

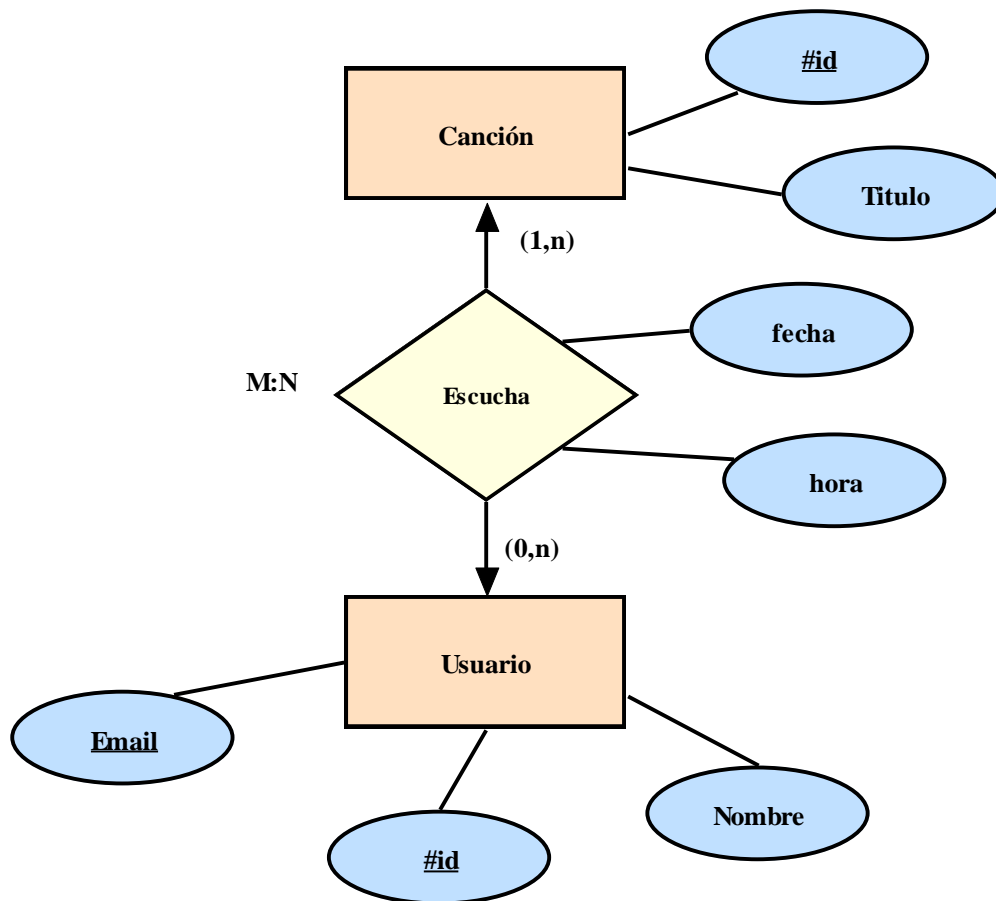


Ilustración 4-7. Relación M:N

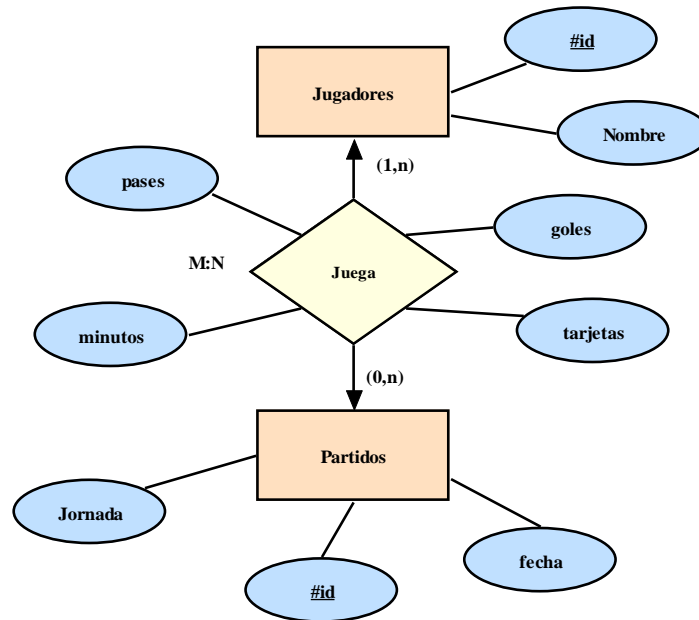
En este ejemplo aplicaremos la regla de transformación de entidades y la regla de transformación de relaciones M:N, así el Modelo Relacional quedaría:

- Canción(#id, Título)
- Usuario(#id, Nombre)
- Escucha(#id_cancion, #id_usuario, Fecha, Hora)
 - Id_cancion -> Cancion(MC, BR)
 - Id_usuairo -> Usuario(MC, BR)

En el caso de que la relación Escucha no tuviese atributos propios:

- Escucha(#id_cancion, #id_usuario)
 - Id_cancion -> Canción(MC, BR)
 - Id_usuairo -> Usuario(MC, BR)

Veamos otro caso práctico



Jugadores(#id, Nombre)

Partidos(#id, fecha, jornada)

Juegan(#id jugador, #id partido, goles, pases, tarjetas, minutos)

Id_jugador -> Jugadores(MC:BR)

Id_partido -> Partidos(MC:BR)

4.4 Transformación de las Relaciones 1:N

A la hora de transformar las relaciones de tipo 1:N al Modelo Relacional habrá que tener en cuenta la cardinalidad mínima de la Entidad que participa con cardinalidad máxima 1 en la relación. Los casos que se pueden dar son los siguientes:

- Que la cardinalidad mínima sea 1 ver figura 4-8
- Que la cardinalidad mínima sea 0 ver figura 4-9

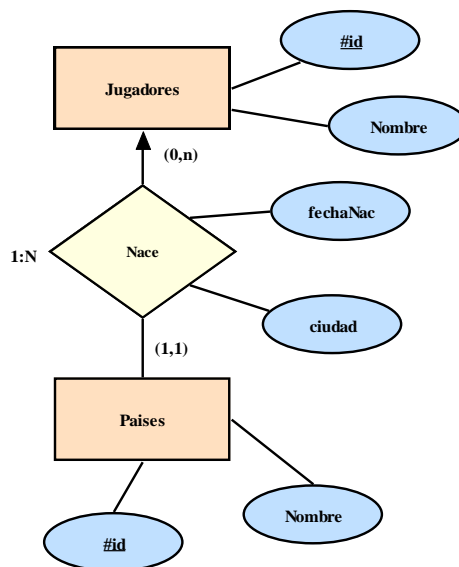


Ilustración 4-8. Cardinalidad mínima 1 de País

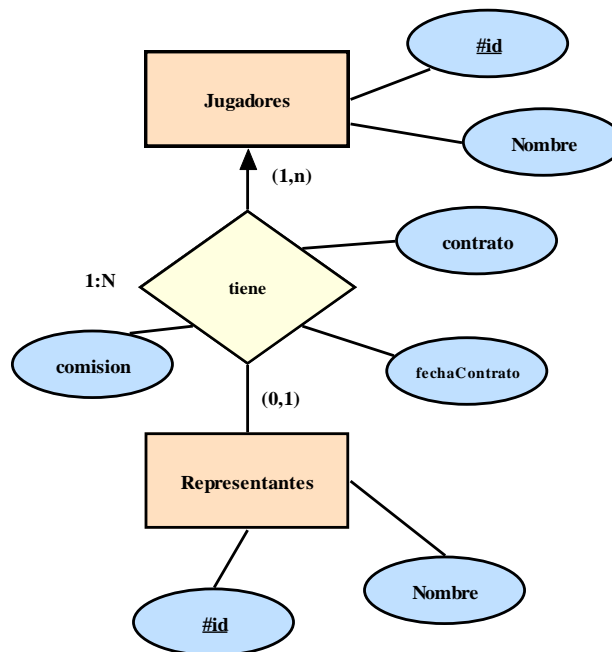


Ilustración 4-9. Cardinalidad mínima 0 de Representante. Es posible que haya jugadores que no tengan representantes.

4.4.1 Caso A. (1,1)

Para el caso en que la entidad que participa con cardinalidad máxima 1 tiene cardinalidad mínima 1, se aplican las siguientes reglas:

- Se propaga la Clave Primaria de la entidad que participa con cardinalidad máxima 1 hacia la que participa con cardinalidad máxima N.
- Debido a la regla anterior la tabla con cardinalidad máxima M dispondrá de un nuevo atributo sobre el que habrá que aplicar la restricción Clave Ajena o FOREIGN KEY, referenciando a la tabla de Cardinalidad 1.
- Si la relación tiene atributos estos se propagan también hacia la entidad que participa con cardinalidad M.

Aplicando las reglas anteriores al ejemplo de la figura 4-8, obtendríamos el siguiente Modelo Relacional:

- Países(#id, Nombre)
- Jugadores (#id, Nombre, FecNacimiento, Ciudad, id_pais)
Id_pais ->Países(MC, BN)

4.4.2 Caso B. (0,1)

Ahora veamos el otro caso en el que la entidad que participa con cardinalidad máxima 1 tiene cardinalidad mínima 0:

- Se crea una **nueva tabla**, cuya clave primaria será la clave primaria de la relación que participa con cardinalidad N, siendo al mismo tiempo Clave Ajena que referenciará a la tabla de donde procede.

- La nueva tabla asumirá también como clave ajena la clave primaria de la tabla que participa con cardinalidad uno.
- En caso de que la relación 1:N tenga atributos han de incluirse además en la nueva tabla.

Aplicando las reglas anteriores a la figura 4-9, obtendríamos el siguiente Modelo Relacional:

- Representante(#id, Nombre)
- Jugadores (#id, Nombre)
- TieneRe (#id_jugador, id_representante, FechaContratacion, Comision, Importe)
 Id_jugador -> Jugadores(MC,BC)
 Id_representante-> Representante(MC, BN)

4.5 Transformación de las Relaciones 1:1

La transformación al Modelo Relacional de las Relaciones de tipo de correspondencia uno a uno, son las más complejas debido a que se pueden dar tres casos distintos:

- Cardinalidad de ambas entidades sea (1,1)
- Cardinalidad de ambas entidades sea (0,1)
- Cardinalidad de una entidad sea (1,1) y de la otra (0,1)

4.5.1 Caso A. E1 (1,1) y E2 (1,1).

Es el caso en el que las dos entidades que participan en la Relación 1:1 lo hacen con cardinalidad (1,1)

En este caso se aplica la siguiente regla:

- Se propaga la clave de cualquiera de ellas a la tabla resultante de la otra, a la cual habría que aplicarle la restricción de Clave Ajena
- La elección de qué clave propagar depende del diseñador y de la semántica de los requisitos.

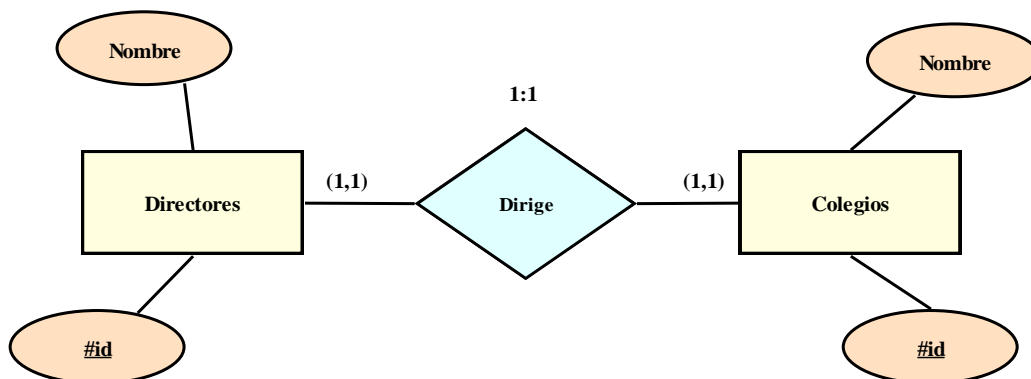


Ilustración 4-10. Caso A

Aplicando la regla anterior el Modelo Relacional resultante sería:

- Directores (**#id**, Nombre)
- Colegios (**#id**, Nombre, id_director)
Id_director->Directores (MC, BN)

Aunque también se podría considerar válida la otra opción:

- Directores (**#id**, Nombre, id_colegio)
Id_colegio -> Colegios(MC, BN)
- Colegios (**#id**, Nombre,)

4.5.2 Caso B. E1(0,1) y E2(0,1)

Las dos entidades que participan en la Relación 1:1 lo hacen con cardinalidad (0,1)

Reglas a aplicar:

- Se crea una **nueva tabla** cuya Clave Primaria sería una de las Claves Primarias de las entidades participantes, la Clave Primaria de la otra entidad pasará a formar parte también de la nueva tabla pero como Clave Ajena.
- En la nueva tabla hay que incluir los atributos propios de la relación 1:1 si los tuviese

Si nos fijamos estas reglas no son equivalentes a las reglas de transformación para las Relaciones M:N, pero sí muy parecidas.

Veamos el ejemplo de la figura 4-11

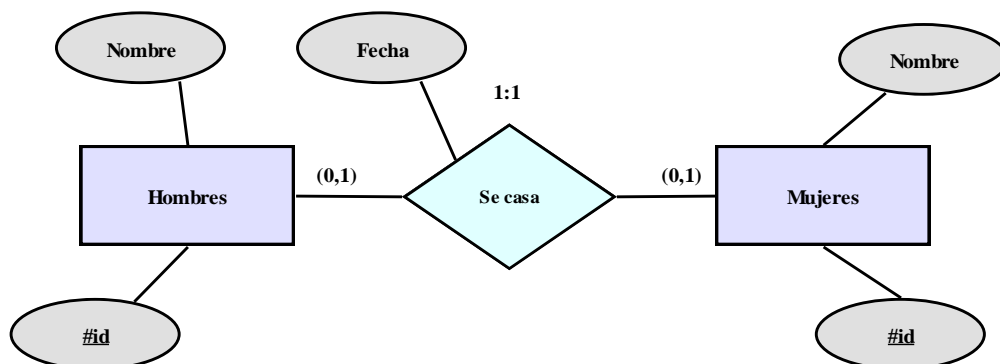


Ilustración 4-11 Caso B

El resultado del Modelo Relacional sería:

- Hombres (**#id**, Nombre)
- Mujeres (**#id**, Nombre)
- EsEsposo (**#id_hombre**, id_mujer, Fecha)
Id_hombre-> Hombres(MC, BC)
Id_mujer-> Mujeres(MC, BC)

4.5.3 Caso C. E1 (0,1) y E2(1,1).

El caso C se corresponde cuando una de las entidades que participa en la relación 1:1 lo hace con cardinalidad (0,1) y la otra con cardinalidad (1,1).

Para estos casos se aplican las siguientes reglas:

- Se propaga la clave de la entidad con cardinalidad (1,1) a la tabla resultante de la entidad que participa con cardinalidad (0,1). De esta forma se evitan valores nulos.
- Si la relación 1:1 tiene atributos, se propagan también hacia la tabla resultante de la Entidad que participa con cardinalidad (0,1)

Veamos el ejemplo de la figura 4-12

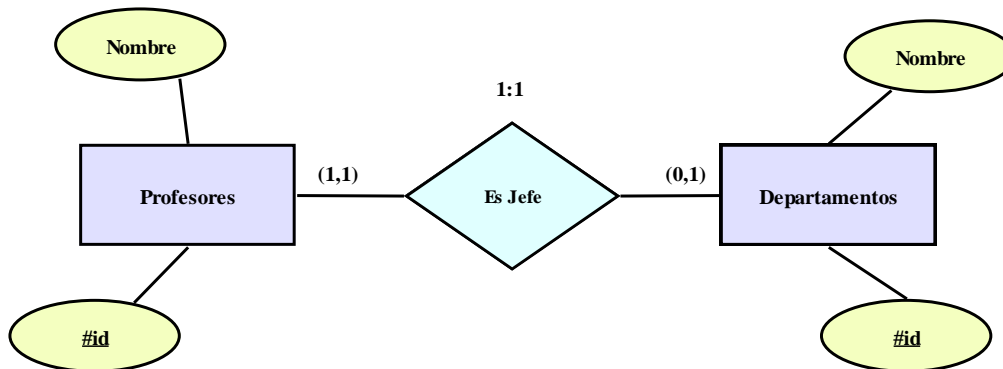


Ilustración 4-12. Caso C

Aplicando las reglas anteriores el Modelo Relacional resultante sería:

- PROFESORES (#id, Nombre)
- DEPARTAMENTO (#id, Nombre, id_profesor)
Id_profesor -> Profesores(MC, BN)

4.6 Transformación de las Relaciones Ternarias

Ya sabemos que una relación ternaria es aquella en la que participan 3 entidades, y que el tipo de correspondencia puede ser:

- 1:1:1
- M:1:1
- M:N:1
- M:N:P

Para este tipo de relaciones aplicaremos la siguiente regla de transformación:

- Se crea una **Nueva Tabla** que tendrá como atributos las Claves Primarias de las tres entidades relacionadas.
- Si la relación tiene atributos serán incluidos también en la nueva tabla.
- La Clave Primaria de la Nueva Tabla dependerá del tipo de correspondencia de la relación:

- M:N:P. La Clave Primaria estará formada por la concatenación de todas las Claves Primarias de las Entidades relacionadas
- M:N:1. La Clave Primaria estará formada por la concatenación de las Claves Primarias de las Entidades que participan con cardinalidad N o M en la relación.
- M:1:1. La Clave Primaria estará formada por la concatenación de la Clave Primaria de la Entidad que participa con cardinalidad M y por la Clave Primaria de cualquiera de las dos entidades que participa con cardinalidad 1.
- 1:1:1. La Clave Primaria estará formada por la concatenación de las Claves Primarias de dos de cualquiera de las tres Entidades relacionadas.

4.6.1 Caso A. M:N:P

Aplicamos las reglas de transformación de las relaciones ternarias pero teniendo en cuenta que:

- La Clave Primaria estará formada por la concatenación de todas las Claves Primarias de las Entidades relacionadas

Veamos el ejemplo de la figura 4-13, en la que deseamos transformar la relación dada en la que un Estudiantes se le asigna una calificación en una Asignatura correspondiente a un determinada Evaluación.

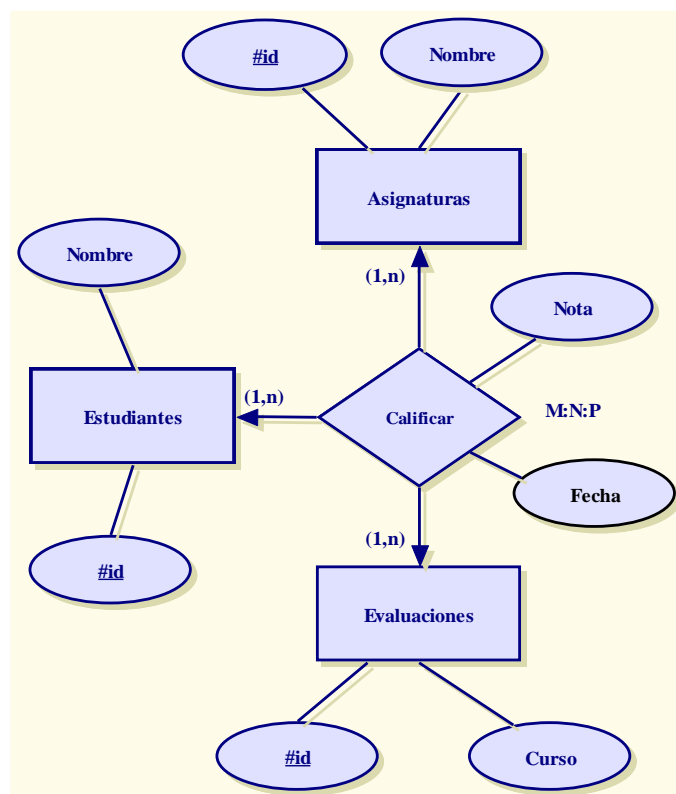


Ilustración 4-13. Relación M:N:P

A partir de la regla de transformación para relaciones M:N:P:

- Estudiantes(#id, Nombre)

- Asignaturas(**#id**, Nombre)
- Evaluaciones(**#id**, Curso)
- Calificar(**#id_estudiante**, **#id_asignatura**, **#id_eval**, Fecha, Nota)
 - Id_estudiante -> Estudiantes(BR,MC)
 - Id_asignatura-> Asignaturas(BR,MC)
 - Id_eval -> Evaluaciones(BR,MC)

4.6.2 Caso B. M:N:1

Aplicamos las reglas de transformación de las relaciones ternarias pero teniendo en cuenta que:

- La Clave Primaria estará formada por la concatenación de las Claves Primarias de las Entidades que participan con cardinalidad N o M en la relación.

Veamos ahora el caso de la figura 4-14 en la que los Toros son lidiados por Toreros en una Plaza

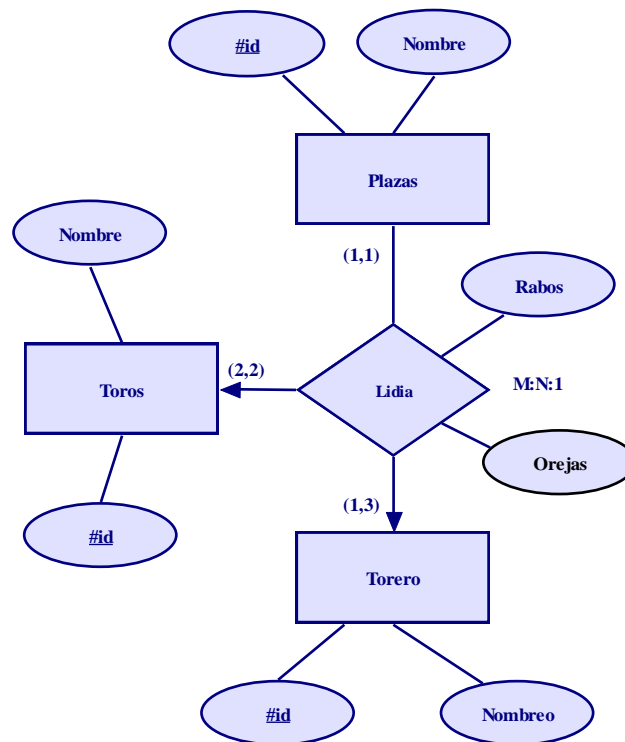


Ilustración 4-14. Relación M:N:1

Aplicando la regla de transformación para las relaciones M:N:1 obtenemos:

- Plazas(**#id**, Nombre)
- Toros(**#id**, Nombre)
- Toreros (**#id**, Nombre)
- Lidia(**#id torero**, **#id toro**, **id_plaza**, Rabos, Orejas)
 - Id_torero -> Toreros(BR,MC)
 - Id_toro -> Toros(BR,MC)
 - Id_palza -> Plazas(BR,MC)

Observamos que no es necesario incluir CodPlaza en la Clave Primaria de Lidia, puesto que un torero sólo va a poder lidiar un toro en una plaza.

4.6.3 CasoC. M:1:1

Aplicamos las reglas de transformación de las relaciones ternarias pero teniendo en cuenta que:

- La Clave Primaria estará formada por la concatenación de la Clave Primaria de la Entidad que participa con cardinalidad M y por la Clave Primaria de cualquiera de las dos entidades que participa con cardinalidad 1.

Veamos el ejemplo de la figura 4-15 donde se representa un modelado necesario para la confección de los horarios en un centro educativo

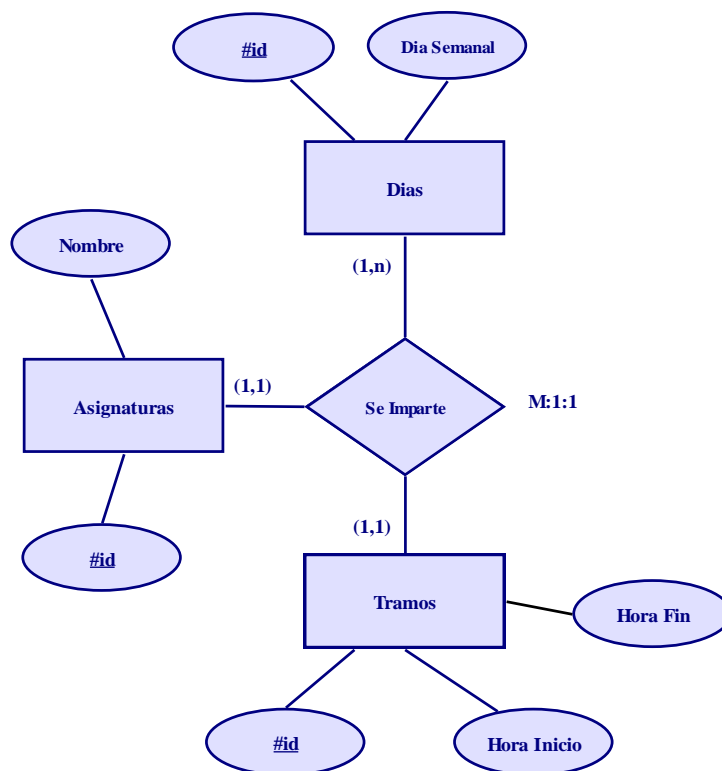


Ilustración 4-15. Relación M:1:1

Aplicando la regla de transformación para las relaciones M:1:1 podríamos obtener dos soluciones igualmente válidas:

Opción A:

- Dias(#id, Dia Semanal)
- Asignaturas (#id, Nombre)
- Tramos(#id, HoraInicio, HoraFin)
- Impartir(#id_dia, #id_asignatura, id_tramo)
 - Id_dia -> Dias(BR, MC)
 - Id_asignatura -> Asignatura(BR,MC)
 - Id_tramo -> Tramos(BR,MC)

Opción B:

- Dias(#id, Dia Semanal)
- Asignaturas (#id, Nombre)
- Tramos(#id, HoraInicio, HoraFin)
- Impartir(#id dia, #id tramo, id asignatura)
 - Id_dia -> Dias(BR, MC)
 - Id_tramo -> Tramos(BR,MC)
 - Id_asignatura -> Asignatura(BR,MC)

4.6.4 Caso D. 1:1:1

Aplicamos las reglas de transformación de las relaciones ternarias pero teniendo en cuenta que:

- La Clave Primaria estará formada por la concatenación de las Claves Primarias de dos de cualquiera de las tres Entidades relacionadas.

Veamos el ejemplo de la figura 4-16

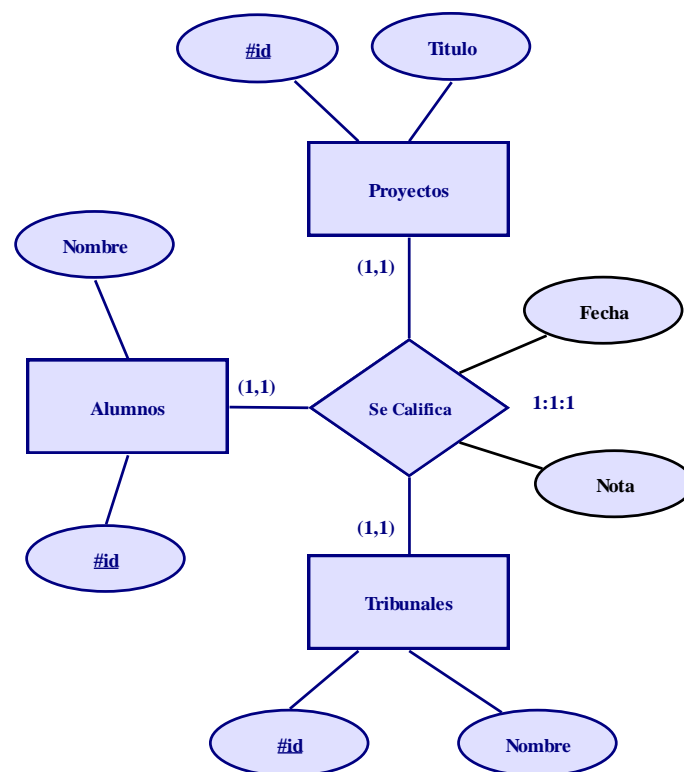


Ilustración 4-16. Relación 1:1:1

Aplicando la regla de transformación para las relaciones 1:1:1 obtenemos:

- Alumnos(#id, Nombre)
- Proyectos(#id, Título)
- Tribunales(#id, Nombre)

- Calificar(#id_alumno, #id_proyecto, id_tribunal, Fecha, Nota)
Id_alumno -> Alumnos(BR,MC)
Id_proyecto -> Proyectos(BR,MC)
Id_tribunal -> Tribunal(BR, MC)

Aunque también serían válidas las siguientes transformaciones para la relación Calificar:

- Calificar(#id_alumno, #id_tribunal, id_proyecto, Fecha, Nota)
- Calificar(#id_proyecto, #id_tribunal, id_alumno, Fecha, Nota)

4.7 Transformación de las Relaciones n-arias

La transformación de las relaciones n-arias se puede ver como una generalización de lo que hemos explicado para las ternarias.

En todos los casos, la transformación de una interrelación n-aria consistirá en la obtención de una nueva relación que contiene todos los atributos que forman las claves primarias de las n entidades relacionadas y todos los atributos de la relación.

Podemos distinguir los casos siguientes:

- a) Si todas las entidades están conectadas con “muchos”, la clave primaria de la nueva relación estará formada por todos los atributos que forman las claves de las n entidades interrelacionadas.
- b) Si una o más entidades están conectadas con “uno”, la clave primaria de la nueva relación estará formada por las claves de $n - 1$ de las entidades interrelacionadas, con la condición de que la entidad, cuya clave no se ha incluido, debe ser una de las que está conectada con “uno”.

4.8 Transformación de las relaciones Reflexivas

Las transformaciones de las relaciones recursivas, reflexivas o unarias son similares a las que hemos visto para el resto de las relaciones.

En general de regla dice:

- Si la relación reflexiva es del tipo 1:1 y las cardinalidades son (0,1) y (0,1) se crea una Nueva Tabla cuya Clave Principal estará formada por la Clave Principal de la entidad participante, la nueva tabla además incluirá una nueva propagación de la clave. Si la relación tiene atributos se incluyen también en la nueva tabla.
- Si la relación reflexiva tiene tipo de correspondencia 1:N o 1:1 en los casos no planteados en la sección anterior, se realiza una propagación de clave desde la entidad relacionada hacia sí misma, igualmente a este nuevo atributo hay que cambiarle el nombre. Si la relación tiene atributos se añaden también a los atributos a la entidad participante.

- Si una relación reflexiva tiene tipo de correspondencia M:N, se crea una Nueva Tabla cuya Clave Primaria estará formada por la concatenación doble de la Clave Primaria de la Entidad participante, como en el modelo relacional no pueden existir dos atributos con el mismo nombre, a una de ellas hay que cambiarle el nombre. Si la relación tiene atributos se incluyen también en la nueva tabla.

Mostraremos la transformación de algunos ejemplos concretos de relaciones recursivas para ilustrar los detalles de la afirmación anterior.

4.8.1 Caso A. 1:1 con (0,1) y (0,1)

Vemos el caso de la figura 4-17

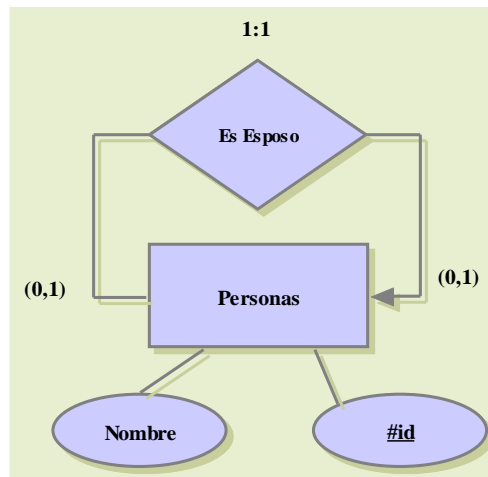


Ilustración 4-17. Relación Reflexiva 1:1

La relación de la figura anterior es recursiva, unaria y tiene conectividad 1:1. Las relaciones 1:1 originan una clave foránea que se pone en la relación que participa. En nuestro ejemplo, la clave foránea deberá estar en la relación PERSONA. Esta clave foránea deberá referenciar a la misma relación para que refleje una relación entre una ocurrencia de persona y otra ocurrencia de persona. Así, obtendremos:

- Personas(#id, Nombre)
- EsEsposo(#id_persona, id_conyugue)
 Id_persona -> Persona(BR,MC)
 Id_conyugue-> Persona(BR,MC)

4.8.2 Caso B. 1:N o 1:1

Veamos ahora el caso de la figura 4-18

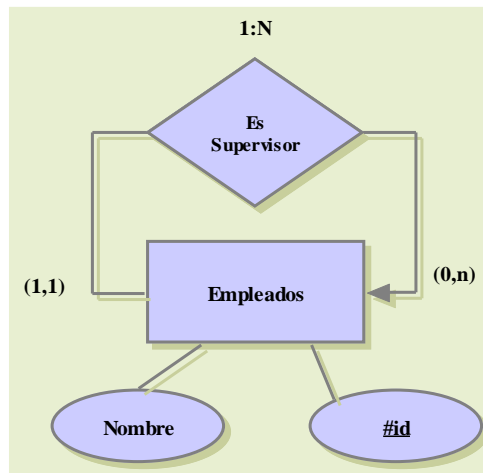


Ilustración 4-18. Reflexiva 1:N

Esta es la norma más común y consiste en propagar la Clave de la Tabla Empleados hacia sí misma, con el obvio cambio de nombre, pasando a ser Clave Ajena que referencia a su misma tabla.

Veamos como quedaría el modelo relacional:

- Empleados(#id, Nombre, id_supervisor)
- Id_supervisor -> Empleados(BR,MC)

4.8.3 Caso C. M:N

Veamos ahora el caso de la figura 4-19

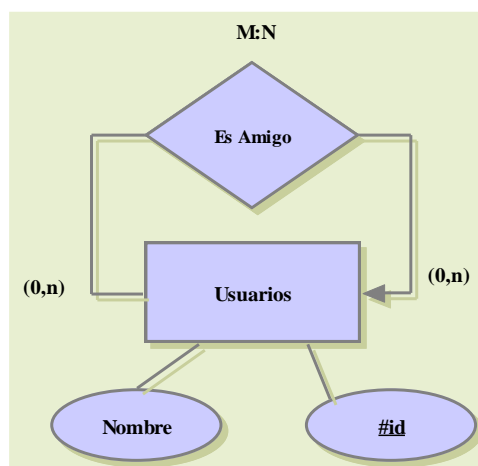


Ilustración 4-19. Reflexiva M:N

En nuestro ejemplo, la interrelación vincula ocurrencias de Usuario con otras ocurrencias de Usuario. En este caso, la clave primaria de la nueva relación estará formada por la clave de la entidad persona dos veces, aunque a alguna de ellas convendrá cambiarle el nombre.

Aplicando la regla anterior para relaciones reflexivas M:N obtendríamos el siguiente modelo:

- Usuarios(#id, Nombre)
- EsAmigo(#id_usuario, #id_amigo)

Id_usuario -> Usuario(BR,MC)

Id_amigo -> Usuarios(BR,MC)

4.9 Transformación de las Relaciones Débiles

La Transformación de las Relaciones Débiles y por consiguiente de las Entidades Débiles son parecidas a lo visto hasta ahora pero con algunas diferencias significativas.

En las relaciones débiles debemos de tener en cuenta a la hora de realizar la transformación el tipo de dependencia:

- **Dependencia en Existencia:** Se realiza una propagación de clave desde la entidad fuerte hacia la entidad débil, además a esa Clave Ajena habrá que ponerle tipo de borrado en cascada obligatoriamente.
- **Dependencia en Identificación.** Se aplica la misma regla anterior, sólo que además la Clave Primaria de la Entidad débil estará formada por la concatenación de su Clave Primaria más la Clave Primaria de la entidad fuerte. Igualmente habrá que aplicar a la Clave Ajena el tipo de borrado en cascada.

4.9.1 Dependencia en Existencia

Veamos el ejemplo de la figura 4-20

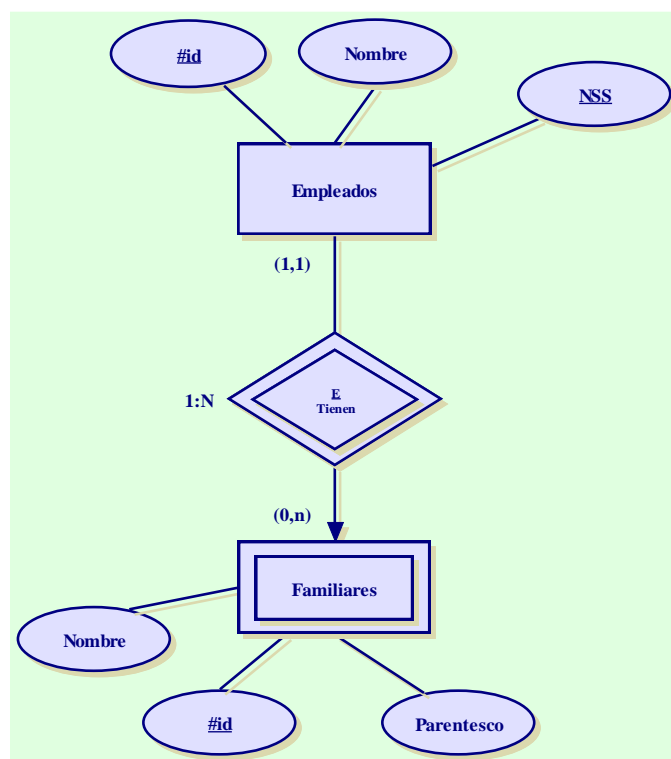


Ilustración 4-20. Dependencia en Existencia

Aplicando las reglas anteriores obtendríamos el siguiente Modelo Relacional:

- Empleados(#id, Nombre, NSS)
- Familiar(#id, Nombre, Parentesco, id_empleado)
- Id_empleado -> Empleado(BC,MC)

4.9.2 Dependencia en Identificación

Veamos ahora el ejemplo de la figura 4-21

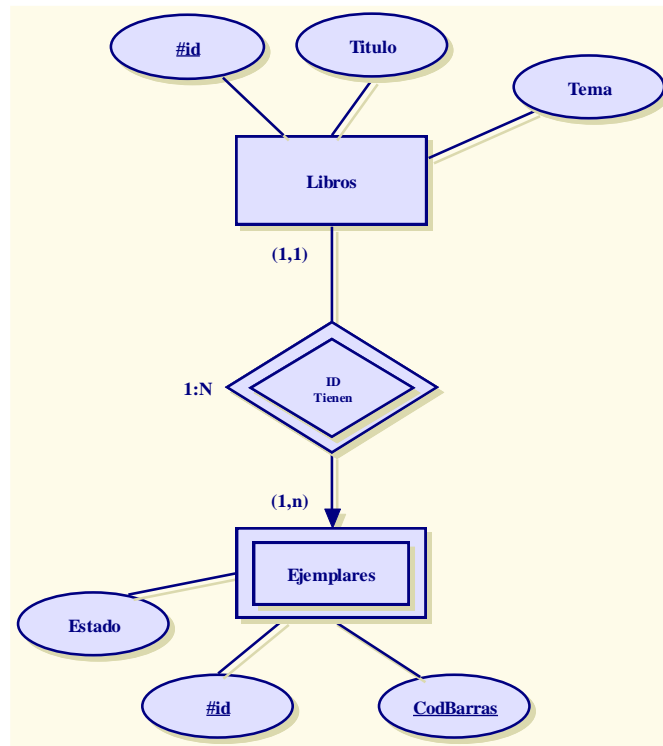


Ilustración 4-21. Dependencia en Identificación

Aplicando las reglas anteriores obtenemos el siguiente modelo:

- Libros (#id, Título, Tema)
- Ejemplares(#id, #id libro, CodBarras, Estado)
- Id_libro -> Libros(BC,MC)

4.10 Transformación de las Jerarquías

Para las jerarquías aplicaremos una regla bastante específica que consiste en lo siguiente:

- Cada una de las entidades supertipo y subtipos que forman parte de una jerarquía se transforman en una relación.
- La relación de la entidad supertipo tiene como clave primaria la clave de la entidad supertipo y contendrá todos los atributos comunes.
- Las relaciones de las entidades subtipos heredan la clave primaria de la entidad supertipo y además contendrán los atributos específicos de los subtipos.
- Las Claves Primarias que heredan los subtipos serán a su vez Claves Ajenas que referenciarán a la tabla resultante del Supertipo.

Apliquemos las reglas al ejemplo de la figura 4-22.

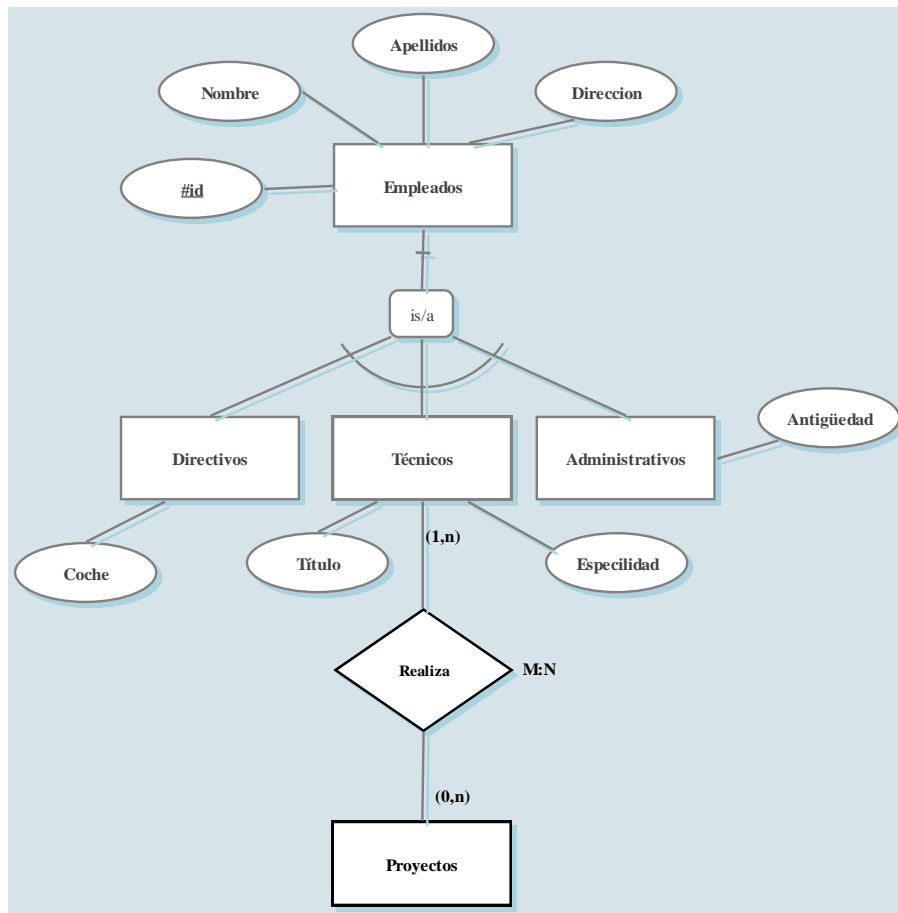


Ilustración 4-22. Jerarquía

Aplicando las reglas anteriores obtendríamos el siguiente modelo:

- Empleados(**#id**, Nombre, Apellidos, Dirección)
- Directivo(**#id_empleado**, Coche)
Id_empleado -> Empleados(BC,MC)
- Técnico(**#id_empleado**, Título, Especialidad)
Id_empleado -> Empleados(BC,MC)
- Administrativo(**#id_empleado**, Antigüedad)
Id_empleado -> Empleado(BC,MC)
- Realiza(**#id_proyecto**, **#id_empleado**)
Id_proyecto -> Proyectos(BR,MC)
Id_empleado -> Tecnicos(BR,MC)