



DEPARTAMENTO
DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Apunte de Modelización

Autores

Sergio D'Arrigo - Leticia Seijas - Cecilia Ruz

Colaborador

Diego A. Castro

Revisiones y Actualizaciones

Alejandro Eidelsztein

1	INTRODUCCIÓN	2
2	DE LOS REQUERIMIENTOS AL MODELO ENTIDAD-RELACIÓN	2
2.1	DESCRIPCIÓN GENERAL	2
2.2	MODELO CONCEPTUAL	2
2.3	MODELO ENTIDAD-RELACIÓN	3
2.4	ELEMENTOS DE UN DER	4
2.4.1	Entidades	4
2.4.2	Atributos	4
2.4.3	Interrelaciones	6
2.4.4	Jerarquías	13
2.4.5	Agregación	15
2.4.6	Restricciones adicionales al modelo	16
3	DEL MODELO ENTIDAD-RELACIÓN AL MODELO RELACIONAL	17
3.1	MODELO LÓGICO RELACIONAL	17
3.2	TRANSFORMACIÓN DEL MER AL MR	18
3.2.1	Entidades Fuertes	18
3.2.2	Entidades Débiles	19
3.2.3	Interrelaciones uno a uno (1:1)	20
3.2.4	Interrelaciones uno a muchos (1:N)	20
3.2.5	Interrelaciones muchos a muchos (N:M)	21
3.2.6	Interrelaciones muchos a muchos (N:M) con atributos	21
3.2.7	Interrelaciones unarias	22
3.2.8	Interrelaciones ternarias	22
3.2.9	Jerarquías	25
3.2.10	Agregación	27
4	BIBLIOGRAFÍA	28

IMPORTANTE:

Este apunte es un complemento de la bibliografía que la materia utiliza como base para el tema. No contiene una visión exhaustiva ni completa de los contenidos, sólo pretende ser un elemento más de ayuda para la comprensión de los mismos.

1 Introducción

El presente apunte está orientado al diseño de Base de Datos Relacionales. Para un sistema cuya persistencia de datos se plantee sobre una base de datos relacional, deberemos definir en tiempo de análisis/diseño las características de esa base de datos. La notación más común para esto es la de Diagramas de Entidad Relación (DER), que es la que utilizaremos en la materia.

El diseño de las bases de datos relacionales puede darse en el contexto de proyectos encarados con diversas metodologías. Si con la metodología utilizada se construye un Modelo Conceptual (MC) con alguna técnica distinta de DER (por ejemplo, los diagramas de clases de UML), éste será uno de los inputs en la construcción del Modelo de Entidad Relación (MER). De no contarse con esto, partiremos de realizar un MC y luego refinarlo de ser necesario para llegar al MER, todo utilizando la notación de DER.

En la materia nos basaremos fundamentalmente en la Metodología de Diseño Lógico para Bases de Datos Relacionales (*LRDM* – Logical Relational Design Methodology), que utiliza la técnica de DER extendido.

En líneas generales, los grandes pasos a realizar para lograr un diseño de la Base de Datos Relacional con esta metodología son:

1. Construcción del Modelo Entidad Relación de los Requerimientos
2. Transformación del Modelo Entidad Relación a Relaciones
3. Normalización de las Relaciones

Con posterioridad al paso 3, se pasará a la construcción física.

En este apunte cubriremos los pasos 1 y 2, que son los que tienen que ver con tareas de modelización.

2 De los Requerimientos al Modelo Entidad-Relación

2.1 Descripción general

El objetivo de este paso es construir un MER que represente los datos a los cuales dar persistencia en la base de datos.

A partir del Modelo de Entidad-Relación, generaremos el Modelo Relacional que servirá para crear la Base de Datos (BD) concreta.

Requerimientos \leftrightarrow MER \leftrightarrow MR \leftrightarrow Normalización \leftrightarrow Diseño Físico \leftrightarrow BD

2.2 Modelo Conceptual

Un modelo conceptual es una conceptualización formal del mundo real (más precisamente de un dominio específico del mundo real). Para construirlo necesitamos modelar las cosas u objetos existentes, sus características y sus relaciones.

Los modelos conceptuales, como todo modelo, son prácticos para comunicar ideas y buscar consensos. La elaboración de un MC es de gran importancia al desarrollar el modelo de análisis de una aplicación y de gran utilidad en la validación con los usuarios.

Para representar un modelo conceptual utilizamos lenguajes de ontologías (lenguajes que nos permiten representar el conocimiento del mundo o parte de él). Estos lenguajes por lo general permiten introducir conceptos, propiedades de los conceptos, relaciones entre los conceptos y algunas restricciones adicionales. Típicamente son expresados por medio de diagramas.

Los Diagramas de Entidad Relación (orientados a Base de Datos Relacionales) y los Diagramas de Clases de UML (orientados al paradigma de objetos), pueden ser considerados lenguajes de ontología.

En Ingeniería de Software I ya se ha trabajado con modelos conceptuales utilizando UML. En Bases de Datos trabajaremos con modelos conceptuales utilizando los Diagramas de Entidad Relación, la técnica más difundida orientada a Bases de Datos Relacionales.

2.3 Modelo Entidad-Relación

El Modelo Entidad-Relación (MER) es una herramienta que permite realizar una abstracción o modelo de alguna situación de interés presente en el mundo real. El MER se realizará utilizando la técnica Diagramas de Entidad Relación (DER). Para construirlo necesitamos modelar las cosas u objetos existentes, sus características y sus relaciones.

El MER se basa en los siguientes elementos:

Entidades

Todo objeto o concepto del cual queremos registrar información constituye una entidad. Las entidades representan conjuntos de elementos¹.

Atributos

Los atributos son propiedades descriptivas de las entidades. Constituyen la información concreta que queremos mantener para cada elemento de una entidad.

Interrelaciones

Las diferentes entidades no están aisladas en el dominio del problema, muchas de ellas van a estar vinculadas entre sí. A esta vinculación la llamamos interrelación. En una interrelación pueden participar una entidad (interrelaciones unarias) o muchas entidades (interrelaciones binarias, ternarias, n-arias en general).

Las interrelaciones unarias son **conjuntos** de pares ordenados de elementos de la misma entidad. Las interrelaciones n-arias ($n > 1$) son conjuntos de n-uplas de elementos de entidades. Notemos que, como son conjuntos, cada tupla de elementos no puede repetirse dentro de una interrelación (lo mismo pasa con los elementos de una entidad).

Reglas de dominio adicionales

Al realizar el MER tenemos la necesidad de modelar ciertas reglas de dominio. En muchos casos el lenguaje que utilizamos no permite expresar a todas ellas. En ese caso, podremos recurrir a ciertas herramientas complementarias que nos permitan expresar ese conocimiento adicional del dominio del problema que no queda expresado en el diagrama.

Se podrá trabajar con lenguaje formal (por ejemplo, OCL) o con lenguaje informal (por ejemplo, lenguaje natural). Lo importante es que haya la menor pérdida de información entre lo conocido relevante del dominio del problema y lo que se exprese con el modelo.

¹ En realidad, siguiendo a alguna bibliografía, estamos llamando entidades a los conjuntos de entidades. Cada entidad es un elemento puntual, pero aquí usaremos indistintamente ambos nombres para representar al conjunto.

En la materia utilizaremos **lenguaje natural** para expresar toda restricción no modelada con DER.

2.4 Elementos de un DER

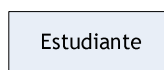
Como dijimos anteriormente, en la materia nos basaremos en la técnica de DER para elaborar modelos. A continuación describiremos la notación de esta técnica y algunas consideraciones del modelado a partir de ella.

2.4.1 Entidades

Todos los conceptos, cosas u objetos del mundo real que queremos modelar constituyen las **entidades** de nuestro modelo.

Las entidades se denotan con un rectángulo, con el nombre dentro de él.

Por ejemplo, la entidad *Estudiante* se representa así:

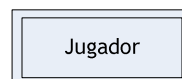


El nombre de la entidad lo escribimos en singular. Debe ser claro y explícito de la información contenida en la entidad, ya que nos da la semántica de la entidad.

Las entidades pueden ser **fuertes** o **débiles**.

Las entidades fuertes son aquellas que tienen una existencia independiente de cualquier otra entidad, se identifican sólo por atributos propios. Las entidades débiles son aquellas que derivan su existencia de otra entidad y necesitan la identificación de dicha entidad para distinguirse de otras (más adelante volveremos sobre las débiles).

Las entidades fuertes se denotan con un rectángulo con línea simple, como el que vimos en el ejemplo anterior. Las entidades débiles se denotan con un rectángulo con línea doble.

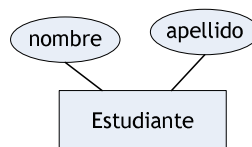


2.4.2 Atributos

2.4.2.1 Descripción General

Los atributos describen a las entidades. Son la información concreta que queremos mantener para una entidad. Se denotan con una elipse vinculada a una entidad, con el nombre dentro de la elipse.

Por ejemplo, si la entidad *Estudiante* tuviera un atributo *apellido* y otro atributo *nombre*, se representaría así:



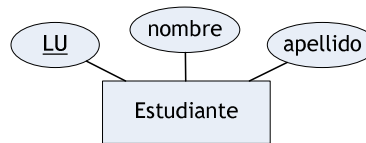
Este tipo de atributos se denominan **atributos simples**. Son simples o atómicos en su estructura (no pueden ser subdivididos en subpartes) y también en sus valores (tienen un solo valor para cada entidad)

2.4.2.2 Atributos identificatorios

Los elementos de cualquier entidad se distinguen por el valor de algún atributo identificatorio (o de varios atributos en conjunto). A este atributo (o conjunto de atributos) se lo denomina **clave primaria** (PK – Primary Key).

La clave primaria es un conjunto minimal de atributos identificatorios y se denota subrayando a los mismos con una línea continua.

Por ejemplo, si el estudiante se identificara por su número de libreta universitaria, se representaría así:



En lo posible hay que tratar de que la clave primaria esté conformada por la menor cantidad de atributos, lo ideal es que sea un único atributo.

■ IMPORTANTE Una característica del MER es que **toda** entidad debe tener indefectiblemente una clave primaria.

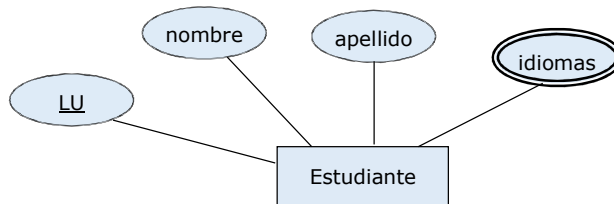
Hay ocasiones donde hay más de un atributo que distingue unívocamente a los elementos de una entidad, en forma independiente uno de otro (por ejemplo, un empleado que se distinga por su número de legajo o por su CUIL). Todos ellos son **claves candidatas**, y se deberá elegir uno de ellos como clave primaria. Las claves candidatas no se modelan en el MER, recién lo veremos al trabajar con el Modelo Relacional.

2.4.2.3 Atributos Multivaluados

La mayoría de los atributos son simples, o sea que tienen un solo valor para cada entidad, como por ejemplo la fecha de nacimiento de un estudiante. Sin embargo, en algunos casos un atributo puede tener un conjunto de valores para una misma entidad, como por ejemplo los idiomas que sabe una persona o los títulos que posee un profesional.

Se los denota con una elipse con línea doble.

Por ejemplo, si la entidad *Estudiante* tuviera un atributo *idiomas*, se representaría así:



2.4.2.4 Atributos Compuestos

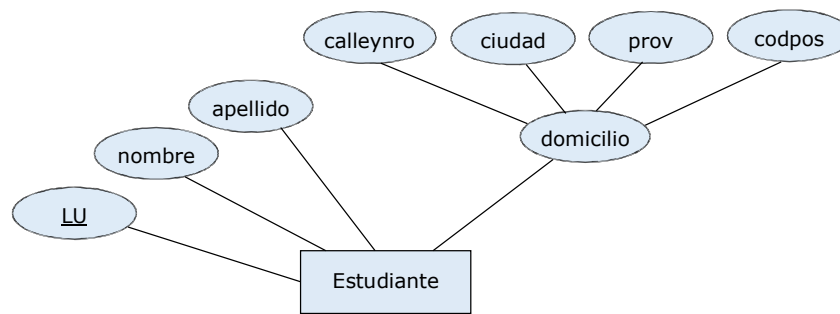
Los atributos compuestos pueden subdividirse en subpartes más pequeñas, las cuales a su vez son atributos con significado independiente. Son útiles para modelar situaciones en las que algunas veces preferimos referirnos al atributo compuesto como una unidad y en otras a sus componentes individualmente.

Los atributos compuestos pueden formar una jerarquía.

El valor de un atributo compuesto es la concatenación de los valores de los atributos simples que lo componen.

Se denotan con una jerarquía de elipses.

Por ejemplo, si la entidad *Estudiante* tuviera un atributo *domicilio* compuesto por *calleynro*, *ciudad*, *prov*, *codpos* se representaría así:



2.4.3 Interrelaciones

2.4.3.1 Descripción general

Una interrelación es una asociación entre entidades. Formalmente una interrelación es una relación matemática sobre $n \geq 2$ entidades (*no necesariamente distintas*).

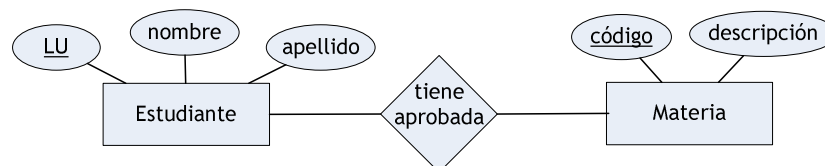
Las interrelaciones se representan con un rombo que vincula las distintas entidades participantes de la misma.

Cada interrelación tiene un nombre, que debe ser lo más representativo posible, ya que ese nombre nos va a indicar la semántica de la interrelación.

El nombre de la interrelación en lo posible debe estar expresado como un predicado (de hecho, una interrelación se puede mapear a un predicado sobre los elementos de las dos entidades).

Por ejemplo, si en nuestro ejemplo quisiéramos representar las materias aprobadas que tiene un estudiante, podríamos agregar una entidad *Materia*, con atributos *código* (PK) y *descripción*, y una interrelación *tiene aprobada* que vincula al estudiante y a las materias aprobadas.

Esta situación se representaría así:



En este caso está claro que la interrelación *tiene aprobada* tiene origen en *Estudiante* y destino en *Materia* (el estudiante *x* tiene aprobada la materia *y*). Por convención, se tratará que las interrelaciones se lean de *izquierda a derecha* o de *arriba hacia abajo*.

En caso de que esto no sea posible o haya ambigüedad en la interpretación de los roles de cada entidad, se escribirá el rol de cada entidad interviniente en la relación junto a la entidad.

Cuando se trata de interrelaciones, hay 3 características a considerar: el **grado**, la **cardinalidad** y la **participación** de las entidades.

En lo que sigue explicaremos la notación de las tres por separado, pero en un modelo deben incluirse todas simultáneamente.

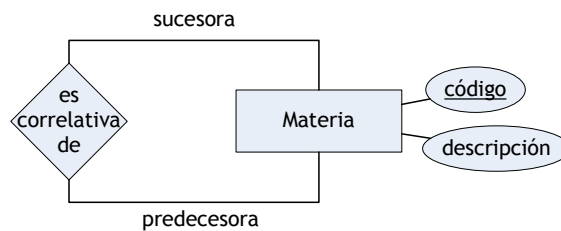
2.4.3.2 Grado

El grado de una interrelación se refiere a la cantidad de entidades que intervienen en ella. En la materia veremos interrelaciones unarias, binarias y ternarias (aunque puede haber de mayor grado, no son frecuentes).

2.4.3.2.1 Interrelación Unaria

En estas interrelaciones participa una única entidad, desempeñando los dos roles de la interrelación. Se la llama también interrelación *reflexiva*. Los elementos de esta entidad se relacionan con elementos de la misma entidad. Notar que a pesar de haber una entidad, constituyen un par ordenado donde cada componente del par desempeña un rol particular en la interrelación.

Un ejemplo de interrelación unaria sería la interrelación *es correlativa de*, que indica las correlatividades entre materias.



■ IMPORTANTE En relaciones unarias siempre se debe aclarar los **roles** de los vínculos para evitar ambigüedad.

2.4.3.2.2 Interrelación Binaria

En esta interrelación participan dos entidades. Los elementos de una entidad se relacionan con los elementos de otra entidad. Las interrelaciones con este grado son las más frecuentes.

Un ejemplo de esta interrelación es la interrelación *tiene aprobada* que vimos anteriormente.

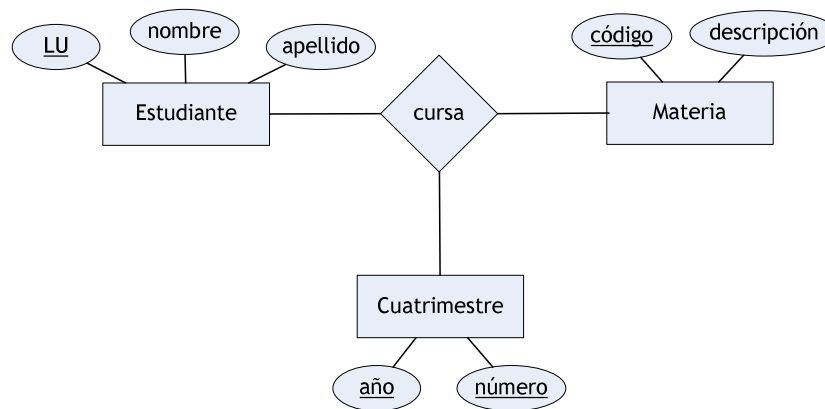
2.4.3.2.3 Interrelación Ternaria

Esta interrelación es más compleja que las anteriores y se da con la participación de tres entidades **simultáneamente**. La interrelación está representada por una 3-upla, donde cada elemento corresponde a un elemento de una de las 3 entidades vinculadas.

Se denota con el mismo rombo, con vínculos hacia las 3 entidades participantes.

■ IMPORTANTE En un elemento de una interrelación ternaria, siempre se requiere la participación de las 3 entidades. Dicho de otra manera, en el marco de estas interrelaciones, no existe la asociación de elementos de dos entidades sin la participación de un elemento de la tercera.

En nuestro ejemplo, si además de tener las entidades *Estudiante*, y *Materia*, tuviésemos también la entidad *Cuatrimestre* (con atributos *año* y *número*), y quisiéramos modelar las materias cursadas por los estudiantes en los diferentes cuatrimestres, podríamos hacerlo con la siguiente relación ternaria:



2.4.3.3 Cardinalidad

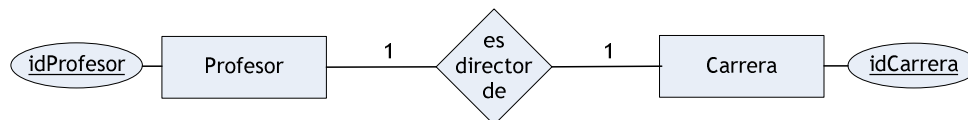
La cardinalidad de una interrelación se refiere a la cantidad de elementos de un rol de la interrelación que puede estar vinculado a un elemento de otro rol de la interrelación.

2.4.3.3.1 Interrelaciones UNO A UNO (1:1)

En las interrelaciones uno a uno (unarias o binarias) los elementos de una entidad se vinculan a lo sumo a un elemento de la otra entidad relacionada.

Se denota escribiendo un 1 junto al vínculo cerca de cada entidad interviniente.

Por ejemplo, si tuviésemos que modelar que cada carrera de la facultad tiene un director que debe ser un profesor, y que cada profesor puede ser director de a lo sumo una carrera, lo podríamos modelar como sigue:



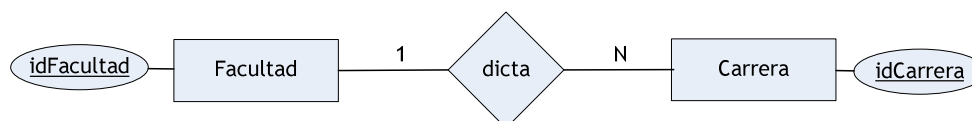
✎ *Notar que el diagrama anterior está incompleto, ya que para ajustarse a la realidad faltaría modelar la noción de participación (que veremos más adelante).*

2.4.3.3.2 Interrelaciones UNO A MUCHOS (1:N)

En las interrelaciones uno a muchos (unarias o binarias), cada elemento de una entidad se puede vincular a muchos elementos de la entidad relacionada, pero los elementos de esta última se vinculan a sólo una instancia de la primera.

Se denota con una N junto al vínculo cerca de la entidad “muchos” y un 1 junto al vínculo cerca de la entidad “uno”.

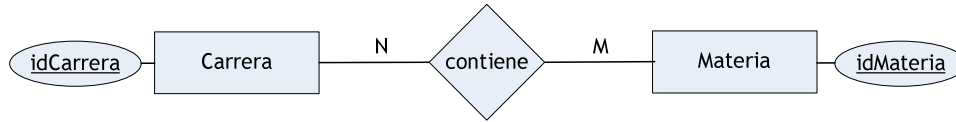
Por ejemplo, si tuviésemos que modelar que cada facultad puede dictar muchas carreras y que cada carrera puede ser dictada por una facultad, lo podríamos modelar como sigue:



2.4.3.3.3 Interrelaciones MUCHOS A MUCHOS (N:M)

En las interrelaciones muchos a muchos (unarias o binarias), cada elemento de cualquiera de las entidades se puede vincular a muchos elementos de la otra entidad relacionada.

Por ejemplo, si tuviésemos que modelar que las carreras pueden contener muchas materias, y que cada materia puede estar contenida en muchas carreras, lo podríamos modelar como sigue:



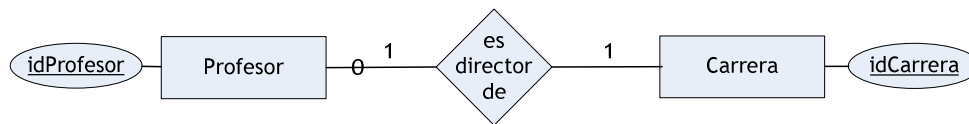
2.4.3.4 Participación de las entidades

Decimos que los elementos de una entidad *participan* de una interrelación. Si todo elemento de la entidad tiene que estar necesariamente en la interrelación (al menos una vez), decimos que la participación es total. Caso contrario (puede haber elementos que no participen) decimos que es parcial u opcional.

2.4.3.4.1 Participación Parcial u Opcional

La participación parcial de una entidad en una interrelación se denota poniendo una "0" sobre el vínculo que une el rombo de la relación con la entidad.

En nuestro ejemplo de los directores de carreras, tenemos que cada profesor puede ser director de a lo sumo una carrera, pero puede no serlo de ninguna. Por lo tanto, la participación de *Profesor* en la interrelación es **parcial**, ya que vamos a poder tener profesores que no estén asociados a ninguna carrera por esta interrelación. Este caso se denotaría así:



Esto anterior podría leerse también como que todo profesor puede dirigir a 0 o 1 carreras.

2.4.3.4.2 Participación Total

En el mismo ejemplo anterior, tenemos que cada carrera tiene que tener al menos un director. Por lo tanto, la participación de *Carrera* en la interrelación es **total**, ya que toda carrera debe estar asociada a algún profesor por esta interrelación.

Este caso se denotaría como está en el diagrama anterior.

2.4.3.5 Atributos de Interrelaciones

Las interrelaciones también pueden tener sus atributos en determinadas circunstancias y estos pueden ser descriptivos o identificatorios.

Los atributos de una interrelación se representan igual que los de las entidades, con la diferencia que están vinculados al rombo de la interrelación.

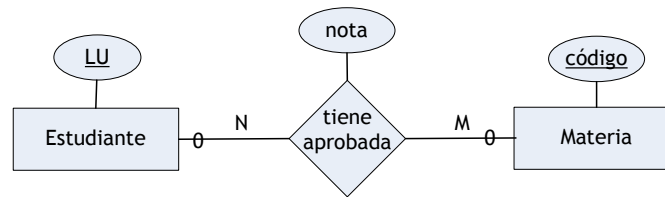
■ IMPORTANTE En nuestra notación, los atributos de interrelaciones (tanto descriptivos como identificatorios) son permitidos **únicamente** en interrelaciones **N:M**.

2.4.3.5.1 Atributos descriptivos de interrelaciones

Un atributo descriptivo de una interrelación permite registrar información adicional a la que aportan de las entidades intervinientes en ella.

Un ejemplo de esto sería si tenemos una interrelación binaria *tiene aprobada* entre *Estudiante* y *Materia* y queremos registrar la nota de aprobación. Este atributo no puede ser de la materia ni del estudiante, ya que depende a la vez de ambas entidades en el contexto de la aprobación. Lo modelaremos como un atributo descriptivo de la interrelación.

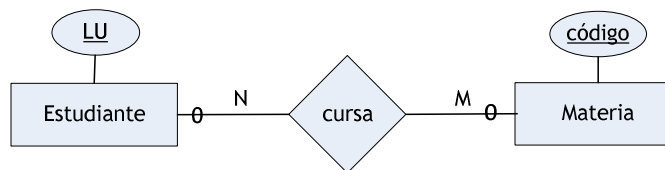
Se denotaría como sigue:



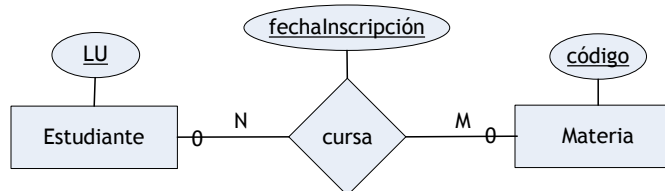
2.4.3.5.2 Atributos identificatorios de interrelaciones

El caso de los atributos identificatorios es un poco más complejo, lo veremos con un ejemplo.

Supongamos que queremos modelar las cursadas como en nuestro ejemplo de ternarias, pero en el dominio de nuestro problema no fuese relevante la entidad *Cuatrimestre* y no se justifique modelarla como entidad. En ese caso la interrelación se daría sólo entre *Estudiante* y *Materia*, constituyendo una interrelación binaria. Esta situación se denotaría como sigue:



Notar que esta interrelación sólo nos permite vincular una vez a cada estudiante con cada materia, ya que los pares de elementos (*Estudiante*, *Materia*) no pueden repetirse (las interrelaciones son un conjunto de pares ordenados). ¿Qué pasaría si quisiéramos reflejar la historia de las cursadas? En este caso necesitamos que esos pares se puedan repetir. La alternativa de notación que podríamos usar en este caso es agregar un atributo identificatorio que nos permita distinguir los elementos de la interrelación con repeticiones, por ejemplo, la fecha de inscripción en la cursada. Se denotaría así:



En este caso, cada elemento de la interrelación estaría identificado por el estudiante (con su *LU*), la materia (con su *código*) y la *fechaInscripción*. Aquí podría repetirse un elemento de estudiante y materia, siempre y cuando tenga diferente fecha de inscripción.

■ IMPORTANTE Notar que los atributos identificatorios sí permiten que se repitan pares ordenados pero éstos deben tener diferentes valores para el atributo identificatorio.

2.4.3.6 Interrelaciones con “muchos” dentro de un rango

Hay casos en que se sabe que la cardinalidad “muchos” está dentro de un rango de elementos (por ejemplo, si dijéramos que cada materia tiene que tener entre 5 y 40 estudiantes inscriptos).

Con la notación que utilizaremos en la materia esta situación **no** se puede modelar.

La forma de expresar este tipo de situaciones es como restricción adicional, en lenguaje natural.

2.4.3.7 Cardinalidad y participación en interrelaciones ternarias

Analicemos la cardinalidad y participación para el caso de interrelaciones n-arias ($n > 2$).

El caso de la participación de entidades en las interrelaciones ternarias es similar al de las binarias. Si todo elemento de una entidad interviniente tiene que estar necesariamente en la interrelación (al menos una vez), decimos que la participación es total. Caso contrario (puede haber elementos que no

participen) decimos que es parcial u opcional. Esto último se denota de la misma manera, con una “O” sobre el vínculo que une al rombo de la interrelación con la entidad en cuestión.

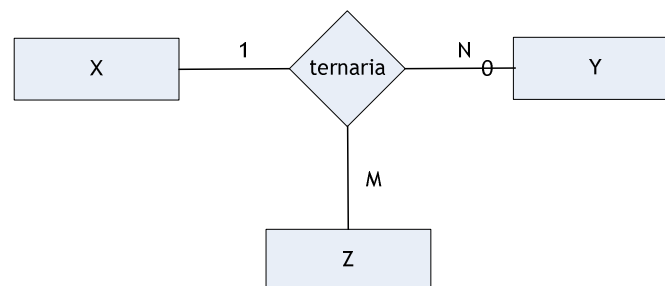
Una interrelación con cardinalidad 1 en una entidad X significa que para cada par de elementos de las entidades restantes que participa de la interrelación, existe un elemento de la entidad X a la que el par está vinculado.

Análogamente, una interrelación con cardinalidad muchos en la entidad X significa que para cada par de elementos de las entidades restantes que participa de la interrelación, pueden existir muchos elementos de la entidad X a las que el par está vinculado.

Notar que el concepto de participación en las interrelaciones está siempre referido a una única entidad. Si en el dominio del problema a modelar existiese alguna restricción de participación que afectase a pares de entidades, esta restricción debería explicitarse como restricción adicional.

2.4.3.8 Interpretación de un caso de interrelaciones ternarias con participación y cardinalidad

Tomemos el siguiente caso de interrelación ternaria.



Se interpretan las siguientes reglas

- Todo elemento de la entidad X debe participar al menos una vez en la interrelación.
- Los elementos de la entidad Y pueden no participar de la interrelación.
- Todo elemento de la entidad Z debe participar al menos una vez en la interrelación.
- Cada par de elementos de las entidades X e Y que participa de la interrelación puede estar vinculado a muchos elementos de la entidad Z.
- Cada par de elementos de las entidades X y Z que participa de la interrelación puede estar vinculado a muchos elementos de la entidad Y.
- Cada par de elementos de las entidades Y y Z que participa de la interrelación está vinculado a un elemento de la entidad X.

Si por ejemplo tuviésemos los siguientes elementos de entidades:

Entidad X: {X1, X2}

Entidad Y: {Y1, Y2, Y3}

Entidad Z: {Z1, Z2}

- Si la interrelación fuese {<X1, Y1, Z1>} sería *inconsistente* con el modelo, ya que se violan las reglas:
 - a: X2 no participa de la relación
 - c: Z2 no participa de la relación
- Si la interrelación fuese {<X1, Y1, Z1>, <X2, Y2, Z2>} sería *consistente* con el modelo.
- Si la interrelación fuese {<X1, Y1, Z1>, <X2, Y2, Z2>, <X1, Y3, Z2>, <X2, Y1, Z1>} sería *inconsistente* con el modelo, ya que se viola la regla:
 - f: El par <Y1, Z1> está vinculado a dos elementos de la entidad X: X1 y X2
- Si la interrelación fuese {<X1, Y1, Z1>, <X2, Y2, Z2>, <X1, Y3, Z2>, <X2, Y3, Z1>} sería *consistente* con el modelo.

La generalización de los conceptos de cardinalidad y participación para interrelaciones n-arias con $n \geq 4$ es más compleja aún y está fuera del alcance de la materia.

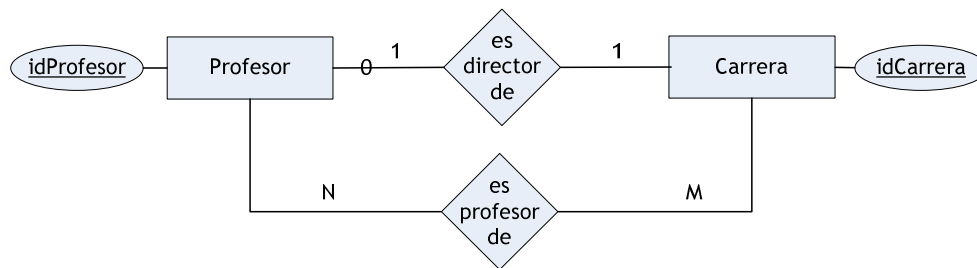
2.4.3.9 Múltiples interrelaciones sobre similares entidades

También puede darse que tengamos más de una interrelación sobre un mismo conjunto de entidades, en este caso cada una de esas interrelaciones tendrá diferente semántica.

En este caso, debemos representar en el modelo las diferentes interrelaciones con rombos independientes, poniendo especial énfasis en los nombres de las mismas, ya que es necesario que su semántica esté bien clara.

Por ejemplo, si tenemos las entidades *Carrera* y *Profesor*, podríamos tener las relaciones *es profesor de* y *es director de* que vinculen a ambas entidades.

Se denotaría así:

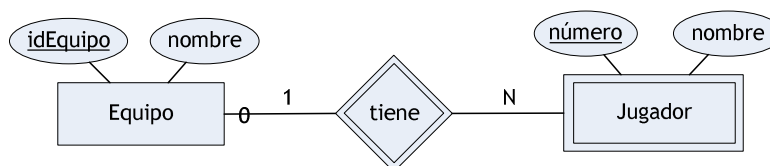


2.4.3.10 Interrelaciones identificatorias de entidades débiles

Como dijimos, las entidades débiles se denotaban con un rectángulo con línea doble. De todos modos, con esto no es suficiente.

Una entidad débil es una entidad que necesita a otra para identificarse. Por lo tanto, tenemos que tener una manera de indicar cuál es la entidad fuerte correspondiente a una entidad débil. Si bien la entidad débil va a estar interrelacionada con su entidad fuerte, también podría estarlo con otras de las cuales no depende su identificación. Para no tener ambigüedades, a la interrelación que vincula una entidad débil con su entidad fuerte correspondiente (llamada *interrelación identificatoria*) la denotaremos también con un rombo con línea doble.

Por ejemplo, si queremos modelar que un equipo puede tener varios jugadores, podríamos modelar las entidades *Equipo* (fuerte) y *Jugador* (débil), con la interrelación *tiene* que las vincula. En este caso, esta interrelación es la interrelación identificatoria.



■ IMPORTANTE Notar que la entidad débil no se identifica por sí sola. El atributo identificador de la entidad débil es parte de la clave y no la clave en su totalidad, y por lo tanto no garantiza unicidad de un elemento. La unicidad va a estar dada por la clave completa de la entidad débil, que está compuesta por la clave de la entidad fuerte (heredada, no se debe explicitar en la entidad débil) más el atributo identificador propio.

■ IMPORTANTE Notar también que, como consecuencia de lo anterior, la entidad débil siempre debe tener participación total en la interrelación identificatoria.

2.4.4 Jerarquías

2.4.4.1 Descripción general

Puede darse que haya entidades que constituyan casos particulares de otras entidades, por ejemplo *Materia con Laboratorio* podría ser un caso especial de *Materia*.

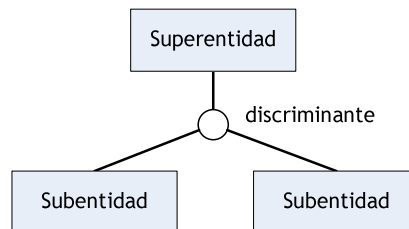
Esta situación la modelamos de una forma especial, con una jerarquía o especialización. Esta relación nos da una jerarquía, una especie de herencia (no es herencia en el sentido completo de objetos).

Con estas relaciones, las propiedades de la entidad padre (la superentidad) son *heredadas* por las entidades hijas (las subentidades).

La identificación de las entidades hijas también es la misma que la de la entidad padre. No puede existir un elemento en una subentidad que no figure en la superentidad.

Cada especialización tiene una semántica y decimos que hay un discriminante que determina la especialización.

La notación está dada por un pequeño círculo vinculado a la superentidad, al cual se vinculan las subentidades, colocando el nombre del discriminante junto al círculo (el discriminante nos indica la semántica de la jerarquía).



En las jerarquías existen 2 características a considerar: la cobertura y el solapamiento.

En lo que sigue explicaremos la notación de las dos por separado, pero en un modelo deben incluirse todas simultáneamente.

2.4.4.2 Jerarquías según cobertura

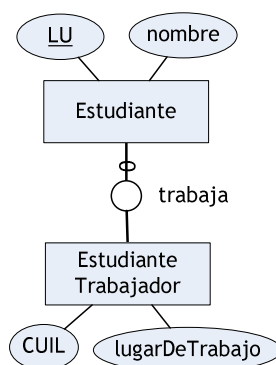
La cobertura nos indica si todos los elementos de la superentidad van a tener algún elemento correspondiente en alguna subentidad.

2.4.4.2.1 Cobertura Parcial u Opcional

En este caso, un elemento de la superentidad no necesariamente está en una subentidad.

Se lo denota colocando una "O" en el vínculo entre el círculo y la superentidad.

Por ejemplo, si quisiéramos registrar alguna información adicional para los estudiantes que trabajan, por ejemplo su CUIL y el lugar de trabajo, lo podríamos representar así:



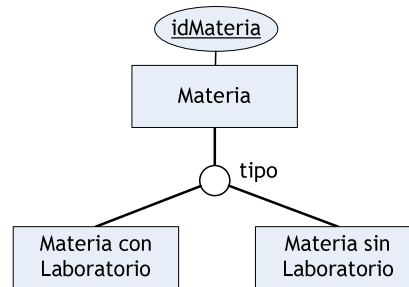
✎ *Notar que el diagrama anterior está incompleto pues falta decir qué tipo de jerarquía es (esto se presentará en breve).*

2.4.4.2 Cobertura Total

En este caso, todo elemento de la superentidad debe estar en al menos una subentidad.

Se lo denota con una línea simple uniendo el círculo con la superentidad.

Por ejemplo, si tenemos que las materias pueden ser con o sin laboratorio (por simplicidad obviamos en este ejemplo los atributos particulares de cada uno), podríamos denotarlo como sigue:



2.4.4.3 Jerarquías según solapamiento

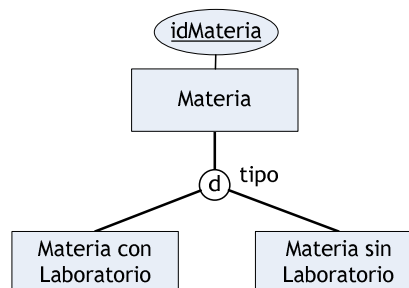
El solapamiento nos indica si los elementos de la superentidad pueden estar en más de una subentidad simultáneamente.

La especialización puede ser disjunta o con solapamiento y se diferencia registrándolo en el círculo de la especialización.

2.4.4.3.1 Jerarquía Disjunta

Los elementos de la superentidad pueden estar a lo sumo en una subentidad. Se lo denota con una **d** en el círculo.

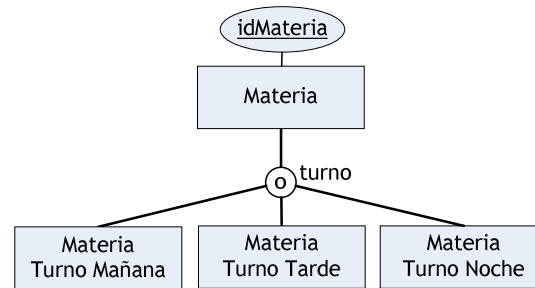
Por ejemplo, con el caso de las materias con y sin laboratorio, quedaría así:



2.4.4.3.2 Jerarquía con Solapamiento

Los elementos de la superentidad pueden estar en más de una subentidad. Se lo denota con una **o** en el círculo (del inglés overlapping).

Por ejemplo, si quisiéramos modelar que una materia puede ser de turno mañana, tarde o noche, pudiendo ser de varios turnos a la vez, lo podríamos modelaríamos así:

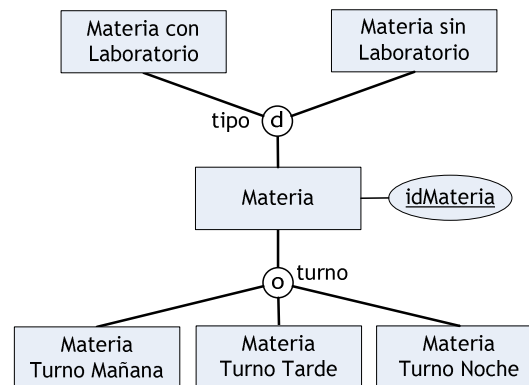


2.4.4.4 Varias jerarquías de una entidad con diferente semántica

Cada jerarquía tiene una semántica. Es posible que una misma entidad tenga diferentes especializaciones con distinta semántica. Se representa vinculando todas las especializaciones a la entidad, escribiendo al lado del círculo el “discriminante” que da la semántica de la entidad.

Por ejemplo, podríamos tener clasificadas nuestras materias según *tipo* (con o sin laboratorio) y también según *turno* que se dictan (mañana, tarde o noche).

Se podría representar así:



2.4.5 Agregación

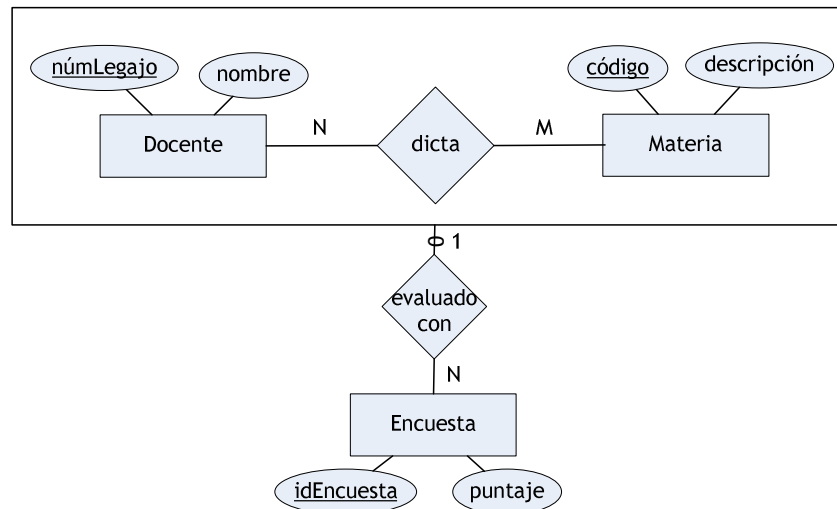
En algunas ocasiones, durante el proceso de modelado surge la necesidad de representar interrelaciones donde participan otras interrelaciones y, por lo que vimos hasta ahora, en nuestro MER esto no está permitido.

Consideremos la siguiente situación: se registran datos de los docentes y de las materias dictadas. Una materia puede ser dictada por muchos docentes y un docente puede dictar muchas materias. Además, en algunos cursos (*no en todos*) se realizan encuestas de evaluación docente. Cada encuesta corresponde a un docente que dicta una materia, y cada docente en una materia puede tener más de una encuesta (o ninguna).

Para modelar este enunciado utilizamos una entidad *Docente*, otra *Materia* y otra *Encuesta* (con las correspondientes claves y atributos descriptivos). Ahora, ¿cómo las relacionamos?

Si pensáramos en una ternaria entre *Docente*, *Materia* y *Encuesta*, estaríamos ante un inconveniente. Como vimos anteriormente, en una tupla de la ternaria **siempre** están los 3 elementos presentes. Por ejemplo, no podemos tener una tupla con 2 elementos y el tercero nulo. Es decir, no podemos tener un docente con una materia sin encuesta (y es justamente lo que queremos modelar).

Aquí es donde aparece en escena la agregación. Entre *Docente* y *Materia* podemos usar una interrelación binaria convencional N:M (*dicta*). Esta interrelación podemos verla como una especie de entidad y relacionarla con *Encuesta* utilizando una binaria (*evaluado con*). Así quedaría representado:



Notar que la binaria *evaluado con* tiene participación parcial del lado de la agregación (esto indica que algunos pares *docente-materia* pueden no tener *encuesta* asociada).

La agregación es una abstracción en la cual una interrelación (junto con sus entidades vinculadas) es tratada como una entidad de alto nivel y puede participar de interrelaciones. Por supuesto, las entidades vinculadas a una agregación también pueden tener sus propias interrelaciones individualmente.

El concepto de agregación se denota con un rectángulo conteniendo a la interrelación y sus entidades vinculadas.

2.4.6 Restricciones adicionales al modelo

Muchas veces, tenemos conocimiento de restricciones que no pueden representarse en nuestro diagrama pero sabemos que constituyen un invariante de nuestro modelo. En este caso, requeriremos que dichas restricciones sean explicitadas en lenguaje natural complementando al modelo.

En nuestro ejemplo de los profesores y directores de los departamentos (sección 2.3.3.9), existiría la siguiente restricción adicional que podríamos enunciar como sigue:

“Todo director de un departamento debe ser profesor de ese departamento”

Es muy común que se requieran restricciones adicionales para interrelaciones “circulares” entre entidades.

3 Del Modelo Entidad-Relación al Modelo Relacional

Una vez que ha sido realizado el *diseño conceptual* de la base de datos, es necesario transformarlo en el *diseño lógico*, dando un paso más hacia la implementación física.

Para expresar el esquema lógico utilizamos un modelo de datos lógico, es decir, un conjunto de herramientas formales que nos permitirán especificar la estructura lógica de la base de datos junto con una descripción de alto nivel de la implementación.

3.1 Modelo Lógico Relacional

El Modelo Lógico Relacional o Modelo Relacional (MR), presentado por Codd en 1970, está ampliamente difundido entre las implementaciones de sistemas de gestión de base de datos. Se basa en una estructura de datos sencilla denominada *relación* y lo sustenta una sólida base teórica.

En este modelo, una base de datos es un conjunto de *relaciones*.

Intuitivamente, una *relación* puede pensarse como una tabla, con filas y columnas. Cada fila, a la que denominaremos *tupla* está formada por un conjunto de valores de datos relacionados que representan un hecho de la realidad. Cada columna de la tabla representa un *atributo*, asociado a un conjunto de valores posibles que puede tomar. A este conjunto lo llamamos *dominio* del atributo.

El nombre asociado a la tabla y cada nombre asociado a una columna o atributo, nos ayudan a interpretar el significado de los valores en cada tupla. El siguiente ejemplo presenta la relación *Empleado*, conformada por atributos *númLegajo*, *nombre*, *apellido*, *fechaNacimiento*, entre otros. La primera fila/tupla nos dice que el empleado con legajo 1102 se llama Martín Gómez y nació el 25 de enero de 1980.

Empleado

<i>númLegajo</i>	<i>nombre</i>	<i>apellido</i>	...	<i>fechaNacimiento</i>
1102	Martín	Gómez		25/01/1980
1105	Natalia	Rodriguez		05/04/1975
1110	Manuel	Pineda		03/10/1979

El conjunto de datos de la relación en un momento dado se denomina *instancia de relación* y la estructura de la relación, es decir su nombre y lista de atributos, se denomina *esquema de relación*.

A continuación describimos los esquemas de relación *Empleado* y *Departamento*, los cuales permiten almacenar los datos de los empleados y departamentos de una organización, además de especificar en qué departamento trabaja cada empleado, si es que está asignado a alguno:

Esquemas de relación

Empleado(númLegajo, nombre, apellido, fechaNacimiento, domicilio, númDepto)

Departamento(númDepto, nombre)

Los atributos subrayados con línea continua son identificadores y constituyen la **clave primaria** de cada esquema.

Los atributos subrayados con línea punteada constituyen una **clave foránea**, es decir, hacen referencia a valores existentes en atributos *clave* de otra relación. En algunas ocasiones que detallaremos más adelante, las claves foráneas pueden tomar valor nulo.

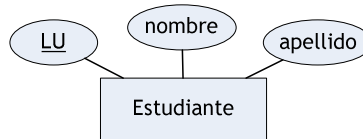
El conjunto de esquemas de relación, junto con las restricciones adicionales asociadas, constituyen el diseño lógico de la base de datos. Es deseable y necesario que el diseño lógico cumpla con ciertas propiedades y no presente anomalías. El proceso de *normalización* es una herramienta que posibilita arribar a un buen diseño de base de datos relacional.

3.2 Transformación del MER al MR

3.2.1 Entidades Fuertes

3.2.1.1 Caso de atributos simples:

Cada conjunto de entidades fuertes de un MER se asocia con un esquema de relación en el modelo relacional, que mantiene el nombre del conjunto de entidades, sus atributos y la clave primaria. A continuación, un ejemplo:

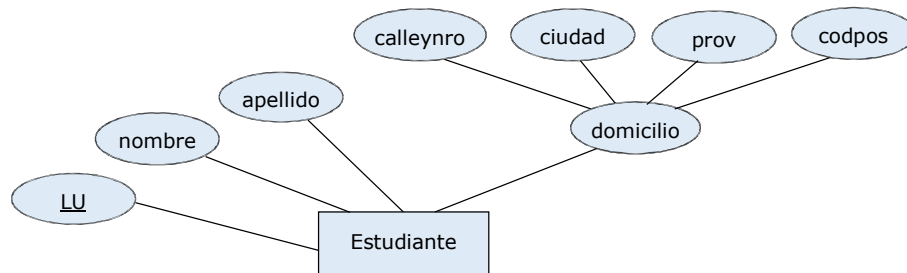


Esquemas resultantes

Estudiante(LU, nombre, apellido)

3.2.1.2 Caso de atributos compuestos:

Cada subparte del atributo compuesto se transforma en un atributo simple. La raíz del atributo compuesto se descarta. Veamos un ejemplo:

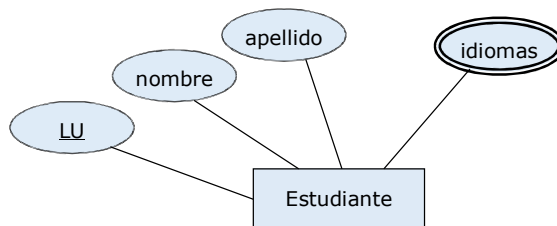


Esquemas resultantes

Estudiante(LU, nombre, apellido, calleynro, ciudad, prov, codpos)

3.2.1.3 Caso de atributos multivaluados:

La regla que se utiliza ahora crea una nueva relación que se identifica por el atributo multivaluado y la clave primaria de la entidad a la que pertenece. Veamos un ejemplo:



Esquemas resultantes

Estudiante(LU, nombre, apellido)

Idiomas(LU, idioma)

Notar que el atributo *LU* en el esquema *Idiomas* hace referencia al atributo con el mismo dominio en *Estudiante*, con lo cual los valores que tome el primero deberán existir en la relación *Estudiante* para mantener la integridad de la base de datos (esto recibe el nombre de **integridad referencial**).

Para cada uno de los esquemas, especificaremos las claves candidatas (CK), primarias (PK) y foráneas (FK) en el caso de existir.

Estudiante PK= CK={LU}

Idiomas PK= CK = {(LU,idioma)}
FK = {LU}

Además, explicitaremos las siguientes **restricciones adicionales**:

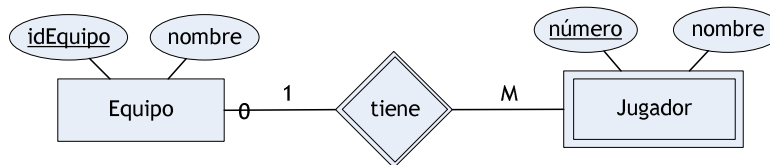
Estudiante.LU puede no estar en *Idiomas.LU*

Idiomas.LU debe estar en *Estudiante.LU*

Notar que escribimos los nombres de los atributos precedidos por el nombre del esquema de relación al que pertenecen.

3.2.2 Entidades Débiles

Una entidad débil se asocia con un esquema de relación en el MR, que mantiene el mismo nombre, sus atributos y presenta como clave primaria la clave primaria de la entidad fuerte asociada junto con la clave parcial de la débil.



Esquemas resultantes

Equipo(idEquipo, nombre)

Jugador(idEquipo, número, nombre)

Notar que el atributo *idEquipo* en el esquema *Jugador* hace referencia al atributo con el mismo dominio en *Equipo*, con lo cual los valores que tome el primero deberán existir en la relación *Equipo* para mantener la integridad de la base de datos.

Para cada uno de los esquemas, especificamos las claves candidatas (CK), primarias (PK) y foráneas (FK) en el caso de existir.

Equipo PK= CK={idEquipo}

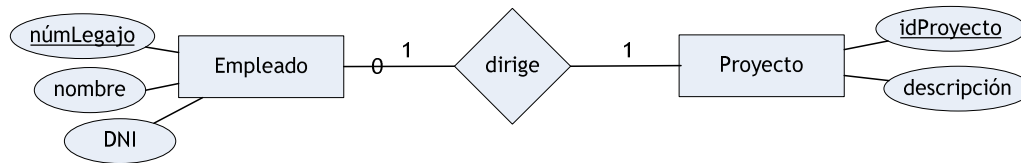
Jugador PK= CK = {(idEquipo,número)}
FK = {idEquipo}

Además, explicitamos las siguientes **restricciones adicionales**:

Equipo.idEquipo puede no estar en *Jugador.idEquipo*

Jugador.idEquipo debe estar en *Equipo.idEquipo*

3.2.3 Interrelaciones uno a uno (1:1)



La regla que se utiliza para la transformación en este caso plantea que se incorpore la clave primaria de una de las entidades como clave foránea de la otra entidad.

Esquemas resultantes

Empleado(númLegajo, nombre, DNI)

PK = {númLegajo}

CK = {númLegajo, DNI}

Proyecto(idProyecto, descripción, númLegajo)

PK = CK = {idProyecto}

FK = {númLegajo}

Notar que el DNI también identifica a un empleado, por lo que es una clave candidata.

Establecemos las siguientes **restricciones adicionales**:

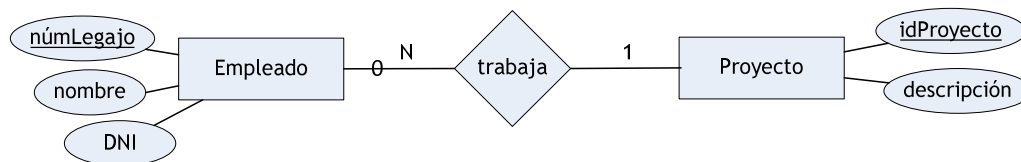
Empleado.númLegajo puede no estar en *Proyecto.númLegajo*

Proyecto.númLegajo debe estar en *Empleado.númLegajo*

■ IMPORTANTE ¿Por qué la FK está del lado del *Proyecto*? Si hubiéramos decidido colocarla en *Empleado*, hubiésemos conseguido una gran cantidad de valores nulos en ese atributo ya que no todos los empleados dirigen un proyecto (pero un proyecto siempre es dirigido por un empleado). En las 1:1 siempre se debe utilizar este criterio para decidir el lugar de la FK.

3.2.4 Interrelaciones uno a muchos (1:N)

Siguiendo con el ejemplo de empleados y proyectos, ahora queremos modelar el hecho de que un empleado puede estar afectado a uno o ningún proyecto, en cambio, un proyecto tiene asignados varios empleados.



La regla que se usa aquí incorpora la clave primaria de la entidad “uno” en la entidad “muchos” como clave foránea.

Esquemas resultantes

Empleado(númLegajo, nombre, DNI, idProyecto)

PK = {númLegajo}

CK = {númLegajo, DNI}

FK = {idProyecto}

Proyecto(idProyecto, descripción)

PK = CK = {idProyecto}

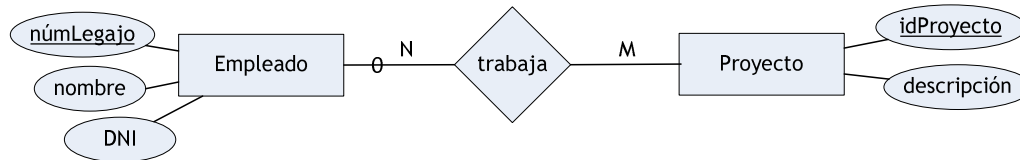
Establecemos las siguientes **restricciones adicionales**:

Empleado.idProyecto es nulo o está en *Proyecto.idProyecto*

Proyecto.idProyecto debe estar en *Empleado.idProyecto*

3.2.5 Interrelaciones muchos a muchos (N:M)

Ahora suponemos que un empleado puede trabajar en varios proyectos a la vez.



La regla que se utiliza ahora crea una nueva relación que se identifica por las claves primarias de las dos entidades interrelacionadas. En este caso no tendrá ningún otro atributo, ya que la interrelación no lo tiene.

Esquemas resultantes

Empleado(númLegajo, nombre, DNI)

PK = {númLegajo}

CK = {númLegajo, DNI}

Proyecto(idProyecto, descripción)

PK = CK = {idProyecto}

Trabaja(númLegajo, idProyecto)

PK = CK = {(númLegajo, idProyecto)}

FK = {númLegajo, idProyecto}

■ IMPORTANTE Notar los paréntesis en la PK de *Trabaja*. Esto significa que la clave es compuesta y la conforman esos atributos al mismo tiempo. En cambio, las FK no cuentan con los paréntesis ya que son 2 simples.

Establecemos las siguientes **restricciones adicionales**:

Empleado.númLegajo puede no estar en *Trabaja.númLegajo*

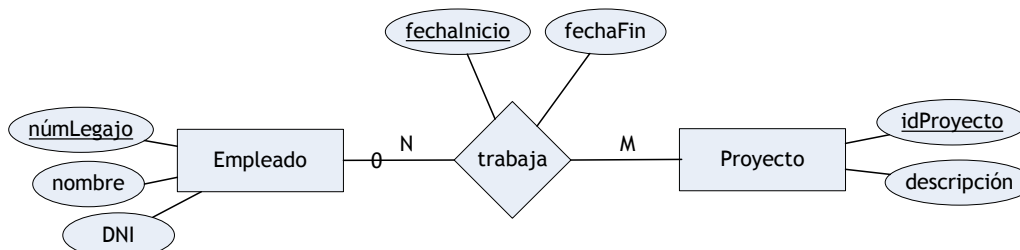
Proyecto.idProyecto debe estar en *Trabaja.idProyecto*

Trabaja.númLegajo debe estar en *Empleado.númLegajo*

Trabaja.idProyecto debe estar en *Proyecto.idProyecto*

3.2.6 Interrelaciones muchos a muchos (N:M) con atributos

Otro ejemplo de transformación de relaciones N:M, es el siguiente: supongamos que tenemos el caso anterior pero queremos modelar la historia de las asignaciones de los empleados en los proyectos.



La regla de transformación que se aplica en este caso es similar a la anterior, con la diferencia que el atributo identificatorio de la interrelación se incorpora a la nueva relación como parte de la clave primaria y el atributo descriptivo de la interrelación como un atributo descriptivo de la nueva relación.

Esquemas resultantes

Empleado(númLegajo, nombre, DNI)

PK = {númLegajo}

CK = {númLegajo, DNI}

Proyecto(idProyecto, descripción)

PK = CK = {idProyecto}

Trabaja(númLegajo, idProyecto, fechaInicio, fechaFin)

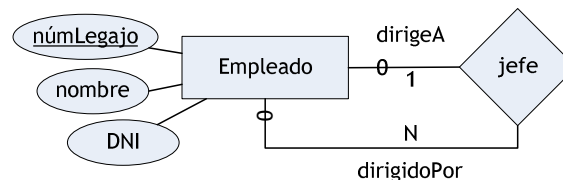
PK = CK = {(númLegajo, idProyecto, fechaInicio)}

FK = {númLegajo, idProyecto}

Las **restricciones adicionales** son las mismas que para el caso N:M.

3.2.7 Interrelaciones unarias

Hasta ahora hemos visto cómo transformar interrelaciones binarias al modelo relacional. Las reglas para transformar interrelaciones unarias son *análogas a las binarias*, lo único que cambia es que ahora tenemos un solo conjunto de entidades para transformar, en lugar de dos.



Esquemas resultantes

Empleado(númLegajo, nombre, DNI, númLegajoJefe)

PK = {númLegajo}

CK = {númLegajo, DNI}

FK = {númLegajoJefe}

Lo anterior nos lleva a estas **restricciones adicionales**:

Empleado.númLegajoJefe puede ser nulo o debe estar en *Empleado.númLegajo*

Empleado.númLegajo puede no estar en *Empleado.númLegajoJefe*

3.2.8 Interrelaciones ternarias

A continuación, presentamos distintos ejemplos de transformación de interrelaciones ternarias al modelo relacional.

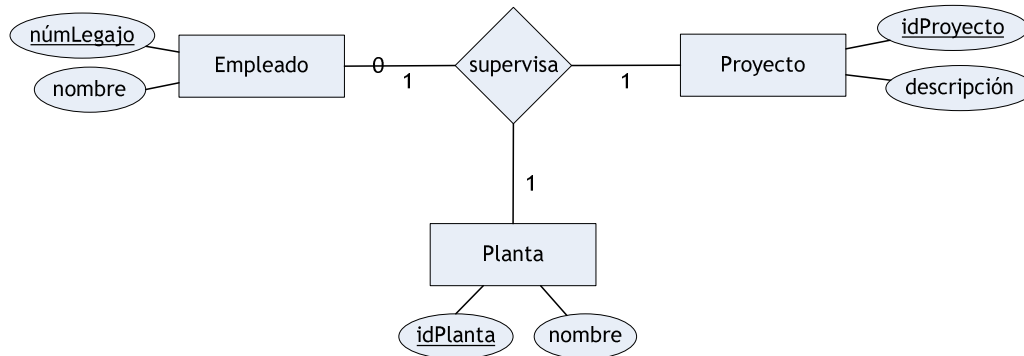
La regla para transformar dichas interrelaciones consiste en crear un esquema de relación por cada conjunto de entidades participante y otro esquema asociado a la interrelación ternaria. Este esquema contendrá los atributos que conforman las claves primarias de las entidades relacionadas; las claves candidatas del mismo serán indicadas en cada ejemplo según la cardinalidad definida.

■ IMPORTANTE Tener en cuenta que las ternarias **siempre** generarán un nuevo esquema de relación. La definición de las claves sí dependerá de la cardinalidad.

3.2.8.1 Interrelaciones ternarias 1:1:1

En una empresa, algunos empleados están encargados de supervisar proyectos que se desarrollan en alguna de las plantas de producción. Un proyecto en una planta está asociado a un supervisor; un

empleado en una planta puede supervisar un proyecto; un empleado que supervisa un proyecto lo hace sólo en una planta. Además, puede haber empleados que no supervisen nada. Modelamos la situación utilizando una interrelación ternaria con cardinalidad 1:1:1.



Esquemas resultantes

Empleado(númLegajo, nombre)

PK = CK = {númLegajo}

Proyecto(idProyecto, descripción)

PK = CK = {idProyecto}

Planta(idPlanta, nombre)

PK = CK = {idPlanta}

Supervisa(númLegajo, idProyecto, idPlanta)

PK = {(númLegajo, idProyecto)}

CK = {(númLegajo, idProyecto), (númLegajo, idPlanta), (idProyecto, idPlanta)}

FK = {númLegajo, idProyecto, idPlanta}

Si analizamos la primera clave candidata, para cada combinación de valores de número de legajo e id de proyecto, podremos determinar el id de planta asociado, que será único. El mismo razonamiento se puede aplicar para el resto de las claves candidatas.

Con algún criterio, que surge del relevamiento del problema, se ha elegido la clave candidata (**númLegajo, idProyecto**) como clave primaria del esquema *Supervisa*.

Establecemos las siguientes **restricciones adicionales**:

Supervisa.númLegajo debe estar en *Empleado.númLegajo*

Supervisa.idProyecto debe estar en *Proyecto.idProyecto*

Supervisa.idPlanta debe estar en *Planta.idPlanta*

Empleado.númLegajo puede no estar en *Supervisa.númLegajo*

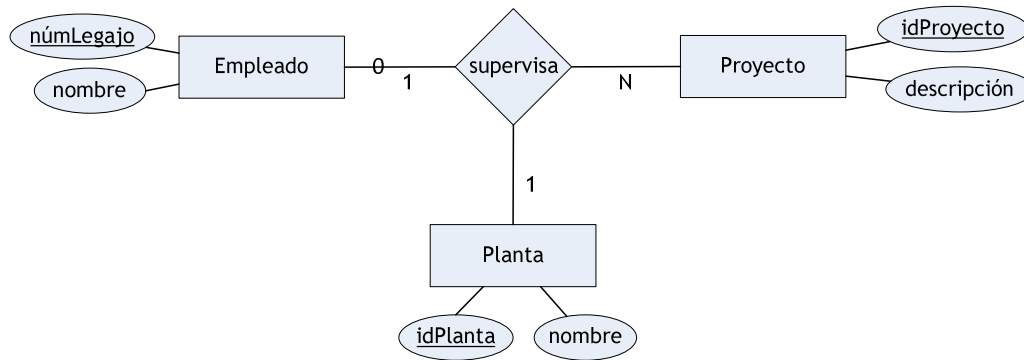
Proyecto.idProyecto debe estar en *Supervisa.idProyecto*

Planta.idPlanta debe estar en *Supervisa.idPlanta*

■ IMPORTANTE Recordar que en toda ternaria, siempre deben estar presentes los 3 integrantes de la ternaria. En este caso, en ninguna de las tuplas de *Supervisa*, el *númLegajo*, *idProyecto* o *idPlanta* podrán tomar valores nulos.

3.2.8.2 Interrelaciones ternarias 1:1:N

Continuando con el ejemplo anterior, ahora un empleado en una planta puede supervisar varios proyectos. Modelamos esta situación utilizando una interrelación ternaria con cardinalidad 1:1:N.



Esquemas resultantes

Empleado(númLegajo, nombre)

PK = CK = {númLegajo}

Proyecto(idProyecto, descripción)

PK = CK = {idProyecto}

Planta(idPlanta, nombre)

PK = CK = {idPlanta}

Supervisa(númLegajo, idProyecto, idPlanta)

PK = {(númLegajo, idProyecto)}

CK = {(númLegajo, idProyecto), (idProyecto, idPlanta)}

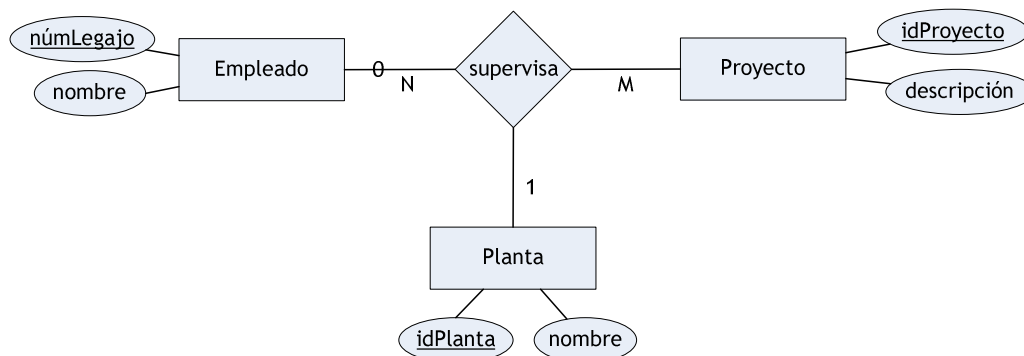
FK = {númLegajo, idProyecto, idPlanta}

Notar que, por la cardinalidad de la interrelación, un valor de número de legajo asociado con un valor de número de proyecto, determinan un único valor de número de planta. De la misma forma, un número de planta asociado a un número de proyecto determinan un único valor de número de legajo de empleado.

Establecemos las mismas **restricciones adicionales** que en el caso anterior.

3.2.8.3 Interrelaciones ternarias 1:N:M

Retomando el ejemplo anterior, ahora un proyecto en una planta puede ser supervisado por más de un empleado. Usamos una ternaria 1:N:M.



Esquemas resultantes

Empleado(númLegajo, nombre)

PK = CK = {númLegajo}

Proyecto(idProyecto, descripción)

PK = CK = {idProyecto}

Planta(idPlanta, nombre)

PK = CK = {idPlanta}

Supervisa(númLegajo, idProyecto, idPlanta)

PK = CK = {(númLegajo, idProyecto)}

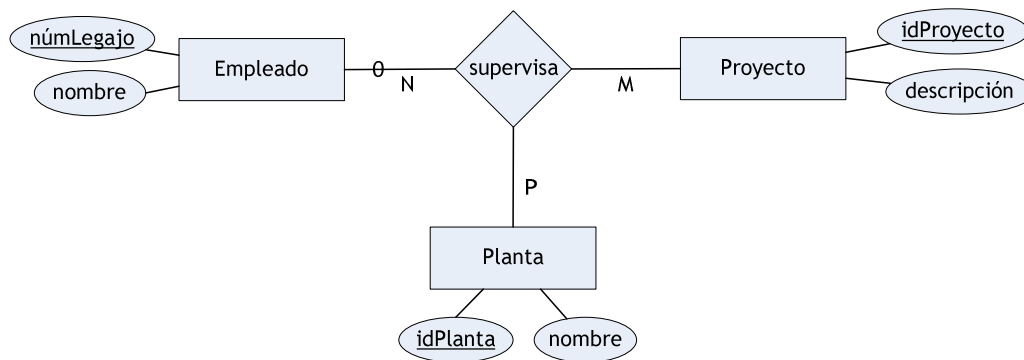
FK = {númLegajo, idProyecto, idPlanta}

A diferencia de los casos anteriores, existe una sola clave candidata. Dado un empleado y un proyecto, la planta queda determinada unívocamente.

Las **restricciones adicionales** son las mismas que en el caso anterior.

3.2.8.4 Interrelaciones ternarias N:M:P

Siguiendo con el ejemplo anterior, ahora un empleado que supervisa un proyecto puede hacerlo en más de una planta. Usamos una ternaria N:M:P.



Esquemas resultantes

Empleado(númLegajo, nombre)

PK = CK = {númLegajo}

Proyecto(idProyecto, descripción)

PK = CK = {idProyecto}

Planta(idPlanta, nombre)

PK = CK = {idPlanta}

Supervisa(númLegajo, idProyecto, idPlanta)

PK = CK = {(númLegajo, idProyecto, idPlanta)}

FK = {númLegajo, idProyecto, idPlanta}

La única clave candidata de *Supervisa* ahora pasa a tener a los 3 participantes, ya que ningún par alcanza para determinar al restante.

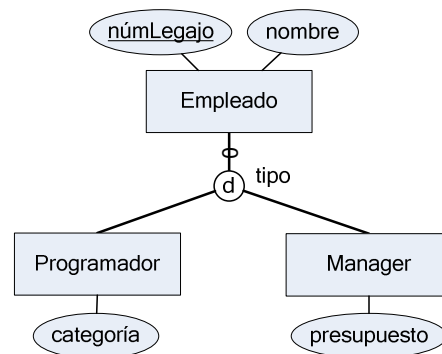
Como antes, las **restricciones adicionales** son las mismas.

3.2.9 Jerarquías

Las reglas de transformación de las jerarquías plantean crear un esquema de relación asociado a la superentidad y un esquema por cada subentidad. Los atributos clave definidos para la superentidad estarán incluidos en los esquemas de relación asociados a cada subentidad, junto con los atributos propios de éstas.

3.2.9.1 Jerarquía Disjunta

Ahora veamos cómo transformar una jerarquía disjunta.



Esquemas resultantes

Empleado(númeroLegajo, nombre, tipo)
 PK = CK = {númeroLegajo}

Programador(númeroLegajo, categoría)
 PK = CK = FK = {númeroLegajo}

Manager(númeroLegajo, presupuesto)
 PK = CK = FK = {númeroLegajo}

■ IMPORTANTE Notar que se agrega el atributo *tipo* en el esquema *Empleado*, que permite particionar el conjunto de empleados. Además, observar que en las subentidades *númeroLegajo* cumple el rol de PK y también de FK.

Establecemos las siguientes **restricciones adicionales**:

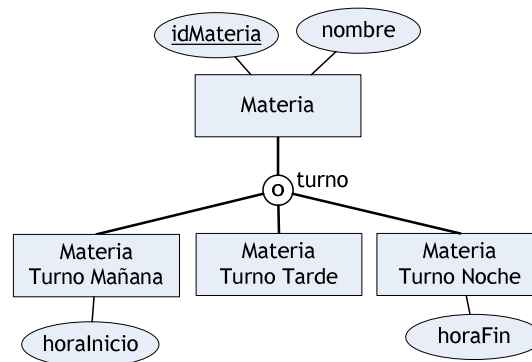
Programador.númeroLegajo debe estar en *Empleado.númeroLegajo*

Manager.númeroLegajo debe estar en *Empleado.númeroLegajo*

Empleado.númeroLegajo puede estar en *Programador.númeroLegajo* o (exclusivo) *Manager.númeroLegajo* o (exclusivo) no estar en ninguno de los dos anteriores.

3.2.9.2 Jerarquía con Solapamiento

Transformemos esta jerarquía sobre materias.



Esquemas resultantes

Materia(idMateria, nombre)
 PK = CK = {idMateria}

MateriaTurnoMañana(idMateria, horaInicio)
 PK = CK = FK = {idMateria}

MateriaTurnoTarde(idMateria)
 PK = CK = FK = {idMateria}

MateriaTurnoNoche(idMateria, horaFin)

PK = CK = FK = {idMateria}

■ **IMPORTANTE** Notar que en el caso de jerarquías con solapamiento, no se agrega un atributo en la superentidad con el nombre del discriminante ya que un elemento de la superentidad puede estar en más de una subentidad. Además, por más que no haya atributos ni interrelaciones en *MateriaTurnoTarde*, es necesario crear un esquema porque sino dejaríamos de modelar a las materias de la tarde.

Agregamos las siguientes **restricciones adicionales**:

MateriaTurnoMañana.idMateria debe estar en *Materia.idMateria*

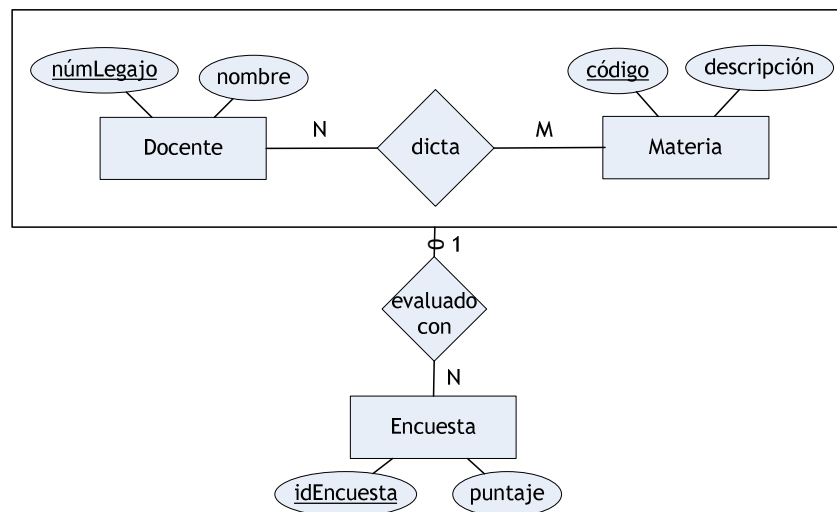
MateriaTurnoTarde.idMateria debe estar en *Materia.idMateria*

MateriaTurnoNoche.idMateria debe estar en *Materia.idMateria*

Materia.idMateria puede estar en *MateriaTurnoMañana.idMateria*, *MateriaTurnoTarde.idMateria* y *MateriaTurnoNoche.idMateria*

3.2.10 Agregación

Las reglas de transformación de la agregación al modelo relacional son parecidas a las que utilizamos para interrelaciones binarias. Se transforman tomando a la *agregación como si fuese una entidad de la binaria*.



Esquemas resultantes

Docente(númLegajo, nombre)

PK = CK = {númLegajo}

Materia(código, descripción)

PK = CK = {código}

Dicta(númLegajo, código)

PK = CK = {(númLegajo, código)}

FK = {númLegajo, código}

Encuesta(idEncuesta, puntaje, númLegajo, código)

PK = CK = {idEncuesta}

FK = {(númLegajo, código)}

■ **IMPORTANTE** Notar que la binaria *evaluado con* se transformó como si participaran dos entidades: *Encuesta* y *Dicta* (claro que esta última no es una entidad sino una agregación). Por este motivo, la FK de *Encuesta* no es hacia *Docente* ni *Materia*, sino directamente hacia la agregación.

Establecemos además las siguientes **restricciones adicionales**:

Dicta.númLegajo debe estar en *Docente.númLegajo*

Dicta.código debe estar en *Materia.código*

Docente.númLegajo debe estar en *Dicta.númLegajo*

Materia.código debe estar en *Dicta.código*

(Encuesta.númLegajo, Encuesta.código) debe estar en *(Dicta.númLegajo, Dicta.código)*

(Dicta.númLegajo, Dicta.código) puede no estar en *(Encuesta.númLegajo, Encuesta.código)*

Como mencionamos previamente, los demás casos de agregación se transforman considerando a la agregación como si fuera una entidad y utilizando las reglas de las binarias.

4 Bibliografía

- Teorey T, Yang D, Fry J – *A Logical Design Methodology for Relational Databases Using the Extended Entity-Relationship Model* – Computing Surveys, Vol. 18, No. 2, 1986
- Elmasri, Navathe - *Fundamentals of Database Systems*, 4th Ed., Addison Wesley, 2003
- Ramakrishnan, Gherke - *Database Management Systems*, 3rd Ed. Mc Graw-Hill, 2003
- Ullman - *Principles of Database and Knowledge Base Systems*, Computer Science Press, 1988
- Chen, P.P.-S., *The entity-relationship model -- toward a unified view of data*. ACM Transactions on Database Systems, 1(1), 9-36. – 1976.
- Enrico Franconi - *Logic – The Meaning of Entity-Relationship Diagrams* – Free University of Bozen-Bolzano.
- Liwu Li, Xin Zhao - *UML Specification of Relational Database* - Journal of Object Technology, vol. 2, N° , September-October 2003, pp. 87-100