

CONTENIDO

1. QUÉ ES MODELAJE?	3
2. UN EJEMPLO SENCILLO	5
3. CONCEPTOS BÁSICOS	8
3.1 Entidades	8
3.2 Relaciones	9
3.3 Atributos	12
3.3.1 Generalidades	12
3.3.2 Identificador Único	14
3.3.3 Atributos opcionales y mandatorios.	15
3.3.4 Representación de atributos.	15
4. EXTENSIÓN AL CASO DE ESTUDIO.	16
4.1 Planteamiento.	16
4.2 Diferentes relaciones definidas sobre las mismas entidades.	18
4.3 Reemplazo de relaciones 1:1 por relaciones 1:N.	18
4.4 Entidades de intersección.	19
4.5 Modelo final.	23
5. IDENTIFICACIÓN DE ENTIDADES, RELACIONES Y ATRIBUTOS.	26
5.1 Introducción.	26
5.2 Pasos del Modelaje.	27
5.2.1 Identificación de entidades.	27
5.2.2 Identificación de relaciones.	28
5.2.3 Elaboración del diagrama entidad/relación.	28

5.2.4 Identificación de atributos.	30
5.2.4.1 Identificador único.	30
5.2.4.2 Número de atributos.	30
5.2.4.3 Caso de estudio.	30
5.3 Supertipos.	32
5.4 Entidades de referencia.	33
5.5 Relaciones condicionales.	36
5.6 Relaciones no transferibles.	37
5.7 Integridad referencial.	38
5.8 Atributos discretos y continuos.	38
6. ESTRUCTURAS CLÁSICAS Y PATRONES GENÉRICOS.	40
6.1 Jerarquías.	40
6.2 Transición de estados	42
6.3 Explosión de materiales.	44
6.4 Clasificación.	45
6.5 Supertipos y clasificación.	46
6.6 Transacciones isomórficas.	47
6.7 Transacciones recurrentes.	50
6.8 Generalización.	51
7. TRANSFORMACIÓN DEL MODELO E/R AL MODELO RELACIONAL (SQL).	52
7.1 Reglas de transformación.	52
7.2 Representación de supertipos.	55
7.3 Representación de arcos.	58
8. APÉNDICES.	62

1. QUÉ ES MODELAJE?

Durante años, se han construido sistemas computarizados cada vez más complejos y de más vasto alcance.

Sin embargo, esta tarea ha sido acometida sin prestar suficiente atención a minimizar la redundancia de los datos, a lograr su verdadera integración y capturar la semántica de los mismos.

El enfoque tradicional se caracterizaba por el desarrollo aislado de aplicaciones, cada una de las cuales poseía su propio sistema de archivos y su propio entorno de operación.

Como consecuencia de ello, los sistemas resultantes estaban aislados entre sí, siendo común que los mismos datos estuvieran almacenados simultáneamente (con distintos niveles de actualización) en diferentes aplicaciones y que resultara sumamente costoso o impracticable integrar datos provenientes de las mismas.

Estos problemas originaron un interés creciente por las tecnologías de bases de datos, interés que hoy se traduce en el uso generalizado de las mismas en proyectos de desarrollo de todas las escalas.

Para lograr un correcto aprovechamiento de la tecnología de bases de datos, no obstante, es imperioso definir un correcto diseño de las mismas, con el propósito de reducir al mínimo la redundancia de los datos y permitir su navegación por todos los conceptos relevantes, especialmente para el soporte a la consulta no programada.

A diferencia de los sistemas de archivos convencionales (en los que las estructuras de datos se modelaban intuitivamente a partir de los requerimientos expresados por el usuario) las bases de datos exigen el uso de técnicas formales para la definición de estructuras de almacenamiento y recuperación.

Así mismo, a diferencia de los programas en lenguajes de tercera generación (en los que la lógica de cómputo se implementaba de una manera informal y algo caprichosa) las herramientas de cuarta generación empleadas con bases de datos suponen el diseño de transacciones basadas en eventos.

El Modelaje es el proceso mediante el cual se identifica las propiedades estáticas (esto es, las estructuras de datos) y dinámicas (esto es, las operaciones) de un dominio de aplicación con miras a su transformación en un diseño implementable en un sistema computarizado.

Así, el Modelaje provee la herramienta conceptual para el analista de sistemas, desde una perspectiva formal, que posibilita la derivación (casi) mecánica de especificaciones de diseño e, incluso, de programación.

Como cabría esperar, no existe una única o mejor forma de Modelaje, pues algunos modelos de datos resultan más apropiados que otros para el análisis de ciertos dominios de aplicación.

Sin embargo, desde hace ya algunos años, se ha estandarizado el uso de una técnica, denominada Modelaje entidad/relación, que resulta apropiada para modelar las propiedades de un amplio espectro de dominios, habiendo sido empleada indistintamente

para el diseño lógico de bases de datos jerárquicas, de redes y relacionales.

Lo que hace al Modelaje entidad/relación tan universal es que no está enfocado al diseño de un modelo de bases de datos particular, disponiendo de técnicas de transformación para muchos de ellos.

El problema central del diseño lógico, como se le concibe hoy día, es el de la correcta y completa representación de propiedades de un dominio de aplicación y el de la transformación de dicha representación en especificaciones de almacenamiento, recuperación y desarrollo de programas de aplicación.

El modelaje entidad/relación se centra en la representación de propiedades del dominio de forma independiente de la implementación y en proveer técnicas de transformación a distintos modelos de bases de datos.

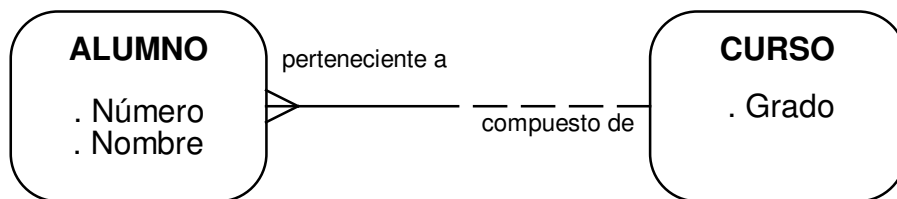
Este documento trata las técnicas básicas del Modelaje entidad/relación y de la transformación del mismo al modelo relacional, más específicamente a bases de datos basadas en SQL, tales como Oracle.

2. UN EJEMPLO SENCILLO

Todos recordamos la organización de los colegios de primaria de nuestra infancia, en los que una misma maestra atendía a todos los alumnos de cada curso, dictando todas la materias del pensum y efectuando evaluaciones periódicas de cada una.

Limitándonos a la perspectiva de los cursos y alumnos, apreciamos que un alumno dado sólo puede estar estudiando en un curso a la vez, en tanto que, en un mismo curso, pueden estudiar uno o más alumnos simultáneamente.

Esta propiedad se ilustra mediante el siguiente diagrama.



Cada una de las cajas del diagrama de la figura representa un entidad y la línea trazada entre las dos una relación definida sobre ellas.

Esta línea tiene un extremo abierto (muchos) al lado izquierdo y un extremo recto (uno) al lado derecho, indicando que en un curso debe haber uno o varios alumnos, y que cada alumno sólo puede estar estudiando en un curso.

La línea continua indica que la relación es mandatoria.

Los ítems anteceditos de un punto son los atributos de cada entidad, siendo propiedades de las mismas que sirven para describirlas.

La relación puede leerse de izquierda a derecha para indicar que:

Cada ALUMNO debe pertenecer a uno y sólo un CURSO

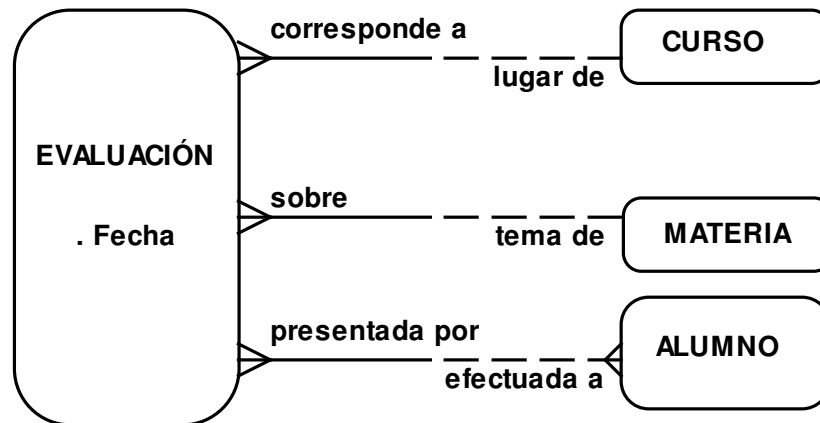
en tanto que, leída de derecha a izquierda, indica que:

Cada CURSO puede componerse de uno o más ALUMNOS.

Nótese que el verbo "debe" se emplea para indicar que la relación es mandatoria.

La relación descrita es una relación 1:N (o "de uno a muchos"). Esta es la forma más común de relación entre dos entidades.

Analizando ahora las evaluaciones que periódicamente se realizan en un curso sobre cada materia, tendríamos un diagrama como este:



En este diagrama apreciamos que la entidad evaluación está asociada simultáneamente con curso, materia y alumno, aunque de maneras diferentes.

Leyendo la línea de relación entre curso y evaluación apreciamos que:

Cada CURSO puede ser lugar de una o más EVALUACIONES

Cada EVALUACIÓN debe corresponder a uno y sólo un CURSO

La línea punteada que parte de curso hacia evaluación indica que en lo que respecta a curso, la relación es opcional (esto es, puede darse que un curso no tenga aún evaluación es como en el primer día de clase).

La línea continua que va desde evaluación a curso indica que en lo que respecta a la evaluación, la relación es mandatoria; esto es, una evaluación sólo puede practicarse en un curso y no tiene sentido fuera del contexto del mismo.

El mismo análisis procede para la relación existente entre evaluación y materia, en la cual la línea de relación puede leerse como:

Cada MATERIA puede ser tema de una o más EVALUACIONES y Cada EVALUACIÓN debe hacerse sobre una y sólo una MATERIA

Examinando la línea de relación tendida entre las entidades evaluación y alumno, nos encontramos con que ambos extremos están abiertos. Esta relación puede leerse como:

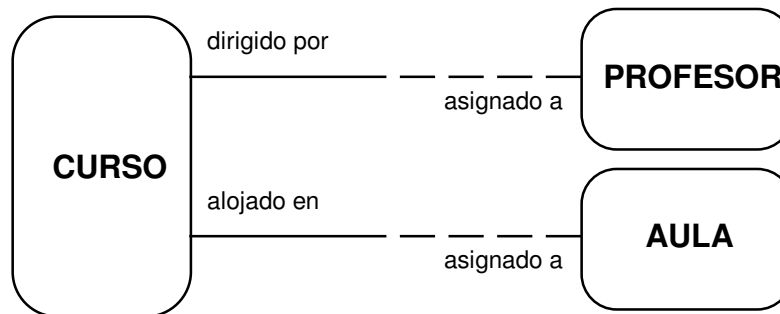
Cada EVALUACIÓN debe efectuarse para uno o más alumnos y Cada ALUMNO puede presentar una o más EVALUACIONES

Nótese que la relación es mandatoria para evaluación y opcional para alumno.

A diferencia de las relaciones anteriores, esta relación es M:N (o de muchos a muchos); esto es, una misma evaluación puede haberse efectuado (simultáneamente) para muchos alumnos, en tanto que un mismo alumno puede haber presentado (en el tiempo) muchas evaluaciones.

Esta es una forma común de relación entre dos entidades, aunque no tanto como la relación 1:N.

Analizando ahora las relaciones entre curso, profesor y aula resulta un diagrama como el siguiente:



Nótese que, en las dos relaciones, ninguno de los extremos está abierto.

La línea de relación tendida entre profesor y curso puede leerse como:

Un CURSO debe estar dirigido por uno y sólo uno un PROFESOR y

Un PROFESOR puede estar asignado a uno y sólo un CURSO

Esto enfatiza el hecho de que no puede existir un curso para el cual no se haya asignado un profesor, en tanto que es razonable suponer la existencia de un profesor al que no se haya asignado un curso.

La línea de relación tendida entre aula y curso puede leerse como:

Un CURSO debe estar localizado en una y sólo un AULA y

Un AULA puede estar asignada a uno y sólo un CURSO

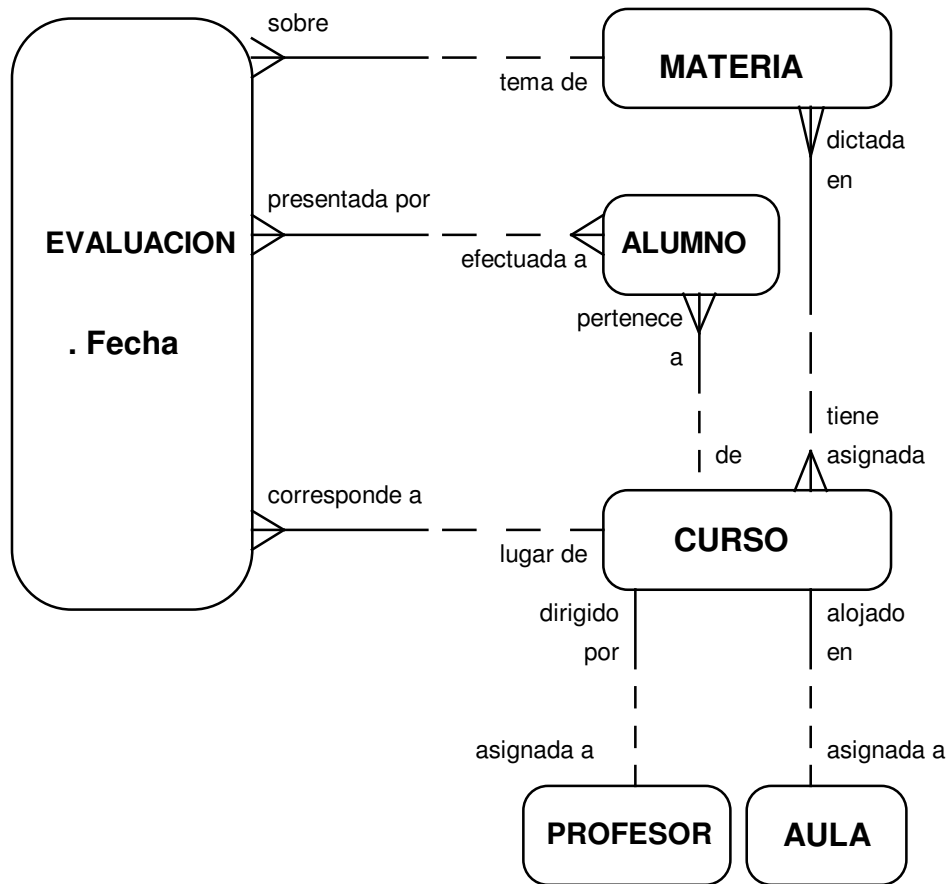
Esta relación establece que no puede existir un curso tal que no esté localizado en un aula, en tanto que un aula puede estar vacía y no haber sido asignada a ningún curso.

En la pagina siguiente se presenta un modelo completo de lo analizado hasta ahora.

Las relaciones (o de uno a uno) son una forma de relación entre dos entidades poco común en la mayoría de los modelos.

Nótese que, por convención, la entidad lado 1 de toda relación es colocada a la derecha o debajo y el lado N a la izquierda o encima.

Nótese también la relación M:N entre Materia y Curso.



3. CONCEPTOS BÁSICOS

En esta sección se definen de manera formal los conceptos básicos del Modelaje entidad/relación: entidad, relación y atributo.

3.1 ENTIDADES

Una entidad es una cosa u objeto significativo (real o imaginario) acerca del cual se requiere conocer o almacenar información.

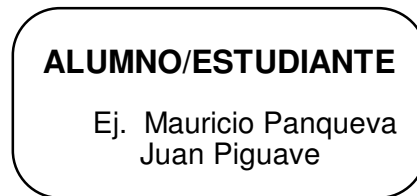
Las entidades se representan diagramáticamente mediante cajas de bordes redondeados, dentro de las cuales se coloca el nombre de la entidad. Este nombre es siempre singular y se presenta en mayúsculas.

El tamaño y disposición de la caja dentro del diagrama son arbitrarios, requiriéndose que posean suficiente espacio para colocar el nombre de la entidad (sin abreviaturas) y dispuestas para hacer el diagrama entidad/relación legible.

Así, es común alargar las cajas para permitir que las líneas de relación se conecten a ellas sin cruzarse o curvarse, con lo que el diagrama parecería una telaraña.

El nombre de la entidad debe representar la clase de objeto tratado, no una instancia. Así, los nombres Albert Einstein o Stephen Hawking no pueden nombrar una entidad; la entidad es CIENTÍFICO y los dos anteriores son instancias de esa entidad.

Es posible que haya sinónimos válidos de uso exactamente equivalente en el dominio de aplicación analizado. Se debe escoger un nombre primario; los sinónimos pueden mostrarse en mayúsculas, precedidos de una barra inclinada / (slash). Los ejemplos pueden mostrarse en mayúsculas y minúsculas.



Toda cosa u objeto debe representarse exactamente mediante una entidad. Esto es, las entidades son mutuamente excluyentes en todos los casos.

Toda entidad debe ser identificable sin ambigüedad. Esto es, toda instancia de una entidad debe ser identificable de forma separada y distinta de todas las demás instancias de la misma entidad. (Véase la sección 3.3.2, Identificador Único).

3.2 RELACIONES

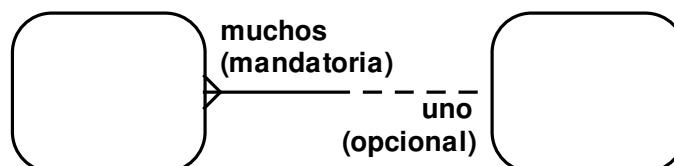
Una relación es una asociación nombrable, significativa y estable entre dos entidades.

Una relación es binaria, en el sentido de que corresponde siempre a la asociación entre exactamente dos entidades o de una entidad consigo misma.

Toda relación tiene dos extremos, para cada uno de los cuales existen:

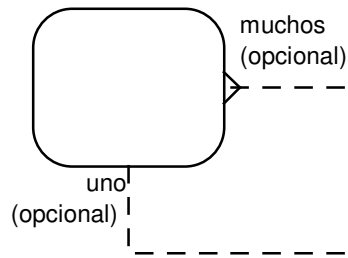
- Una leyenda.
- Un grado o cardinalidad (uno o muchos).
- Una opcionalidad (opcional o mandatoria).

Una relación se representa mediante una línea que conecta las cajas correspondientes a las dos entidades o que conecte recursivamente a una caja consigo misma.



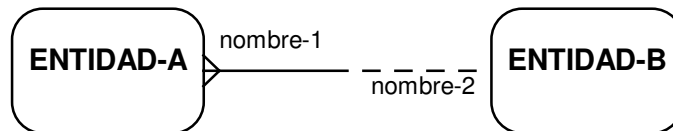
La relación más común es la de grado 1:N, mandatoria en el extremo N y opcional en el extremo 1.

Una relación recursiva es una jerarquía definida sobre una misma entidad, como se muestra en el diagrama.



Este diagrama podría corresponder, por ejemplo, a la jerarquía de cuentas de un plan contable.

El nombre de cada relación se coloca en minúscula junto al extremo apropiado, como se muestra.



Cuando el extremo de la relación es mandatoria se emplea el verbo debe antes del nombre de la relación. Para relaciones opcionales, se emplea el verbo puede.

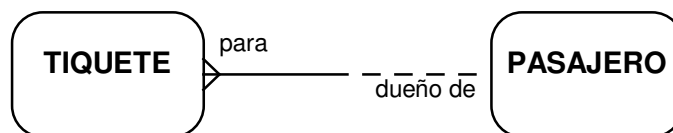
Así, el diagrama anterior se lee de izquierda a derecha como:

Cada ENTIDAD-A debe ser nombre-1 uno y sólo un ENTIDAD-B

y leído de derecha a izquierda:

Cada ENTIDAD-B puede ser nombre-2 uno o más ENTIDAD-A

Esto puede parecer incomprensible hasta que se lea un ejemplo real.



Cada TIQUETE debe ser para uno y sólo un PASAJERO y

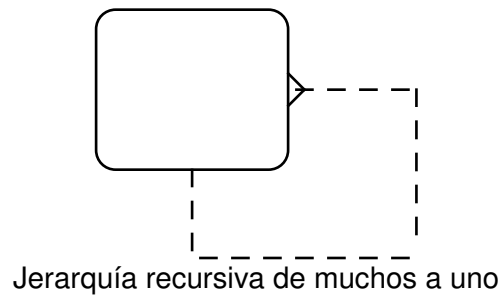
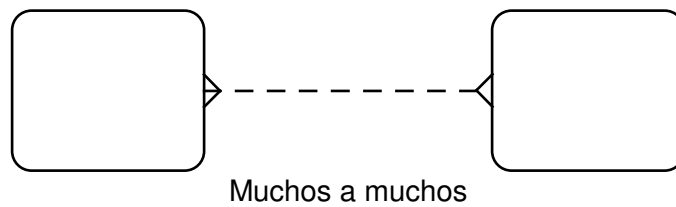
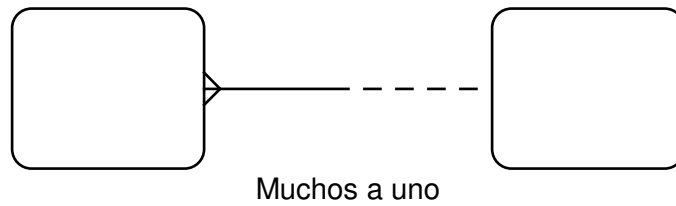
Cada PASAJERO puede ser dueño de uno o más TIQUETES

El plural del nombre de la entidad se emplea cuando el grado es muchos. El grado de muchos se lee como uno o más. El grado uno se lee como uno y sólo uno.

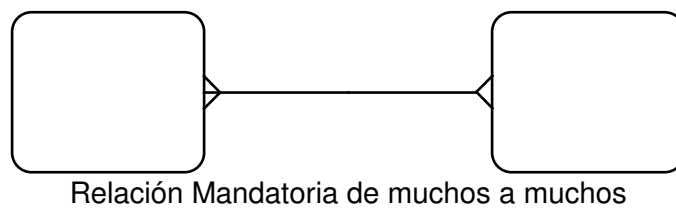
Al elaborar diagramas entidad/relación se logra mayor exactitud al colocar el extremo

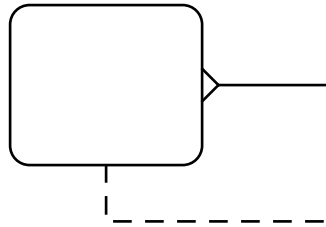
abierto (muchos) en el lado izquierdo o superior. Adicionalmente, el uso de los verbos ser y estar provee nombres de la relación mas significativos y útiles.

No todas las posibles combinaciones de opcionalidad y grado son válidas o frecuentes.



Las relaciones anteriores son muy comunes. Las relaciones siguientes son inválidas, pues representan condiciones imposibles.





Jerarquía infinita

3.3 ATRIBUTOS

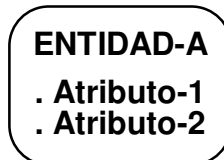
3.3.1 Generalidades

Un atributo es cualquier detalle que sirve para:

- Identificar.
- Describir.
- Cualificar.
- Clasificar.
- Expresar el estado de una entidad.

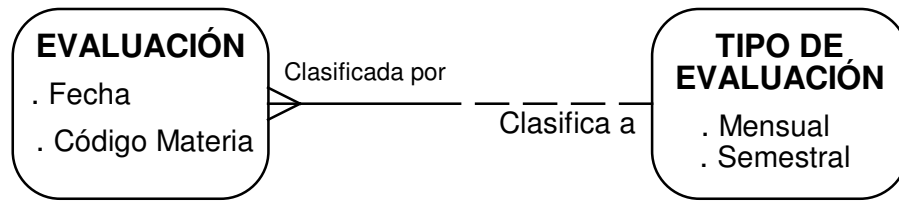
Un atributo puede ser texto, un número, una figura y así sucesivamente, según se requiera. En procesamiento de datos se tiende a utilizar únicamente texto y números, pero es razonable incluir otros tipos de datos tales como gráficos (por ejemplo, fotografías) o sonido.

Para representar un atributo, se escribe su nombre en minúscula y singular, opcionalmente acompañado de un ejemplo de su valor.



En un diagrama entidad/relación no es necesario incluir todos los atributos de cada entidad, pero añadir uno o dos durante el periodo inicial del Modelaje es muy benéfico. En particular, es útil para distinguir entidades tipo de entidades instancia.

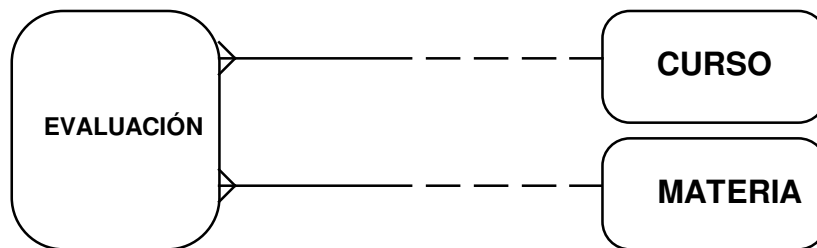
En el ejemplo de la siguiente figura, los atributos resultan necesarios para distinguir las dos entidades.



En esta caso, puede haber sólo cuatro o cinco tipos de evaluación, de acuerdo con su periodicidad, pero puede haber cientos de evaluaciones a lo largo del año escolar.

Los atributos deben describir únicamente a las entidades con las que están asociadas. Esto puede parecer obvio, pero es el error más común asociado con la identificación de atributos. Por ejemplo, es 'nombre de materia' un atributo de evaluación o de materia propiamente dicho?. Obviamente, es un atributo de materia, pero en el mundo real lo vemos replicado en muchos contextos, incluyendo la evaluación.

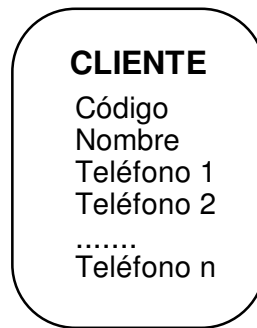
Por qué?. En un sistema manual, usar el nombre de materia es una forma muy conveniente de representar una relación. Cuando se encuentran estas situaciones, se debe trazar una línea de relación, creando una nueva entidad si es necesario, como se ilustra en la figura.



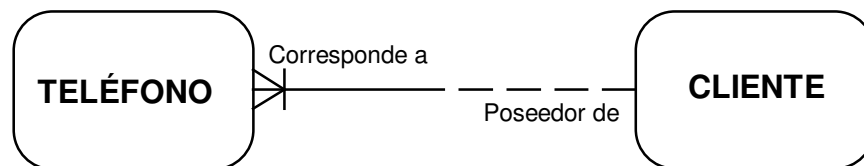
Como guía, una entidad sólo requiere ser descrita por entre dos y ocho atributos. Si se encuentran más de ocho, probablemente hay entidades o relaciones ocultas en el modelo.

No se debe emplear el nombre de la entidad como parte del nombre del atributo, pues resulta redundante, ya que un atributo describe únicamente a una entidad.

Una entidad sólo puede tener un valor para cada atributo en un momento dado. Si resulta esencial tener múltiples valores, es necesario crear una nueva entidad para almacenarlos y una relación de muchos a uno con la entidad original, como se ilustra.



Siguiendo la regla anterior, tendríamos el diagrama de la figura siguiente.



Esta regla corresponde a la llamada primera forma normal.

El nombre de un atributo debe escribirse siempre en singular y minúscula. Un nombre plural coincide, generalmente, con el problema de repetición mencionado anteriormente.

Así mismo, la repetición de atributos puede revelar la existencia de entidades faltantes en el modelo.

Un atributo se transforma en una entidad cuando tiene significado completo en sí mismo, con relaciones y atributos propios.

Este es el caso del atributo Tipo Evaluación en la entidad EVALUACIÓN, que tiene como atributo propio la periodicidad en número de meses y debe, por tanto, ser tratado como una entidad en sí misma.

3.3.2 Identificador Único

Toda entidad debe poder ser identificada con unicidad mediante uno de sus atributos o una combinación de los mismos, denominado clave primaria o identificador único. Por tanto, siempre resulta necesario determinar qué atributo o atributos sirven para identificar una entidad.

Así mismo, todos los atributos de una entidad deben depender únicamente del valor de la clave primaria.

Si hay atributos que dependen sólo de parte de la clave primaria, deben ser removidos y trasladados a una nueva entidad. Esta operación conduce a la llamada segunda forma normal. Como ejemplo de ello tenemos la pregunta:

Depende el nombre de la materia del número de evaluación?

Obviamente no, el nombre de la materia no cambia cuando cambia el número de la evaluación y, por tanto, corresponde a una entidad separada, en este caso, MATERIA.

3.3.3 Atributos opcionales y mandatorios.

Un atributo puede tener valor sólo durante una parte del tiempo o ser desconocido. Esta clase de atributos se denomina opcional y se representa mediante un pequeño círculo antepuesto al nombre del atributo.

El valor de un atributo que debe ser siempre conocido se representa mediante un asterisco (*) antepuesto al nombre del atributo.

Nótese que esto se aplica a los atributos cuyo valor debe ser siempre conocido, invalidando instancias de la entidad en que no se halla valor para estos.

3.3.4 Representación de atributos.

Toda entidad debe ser identificable en forma única, de manera que cada instancia de la misma se pueda diferenciar de modo separado de las demás instancias de la misma entidad.

Los atributos empleados para diferenciar instancias conforman el identificador (o clave primaria) de la entidad.

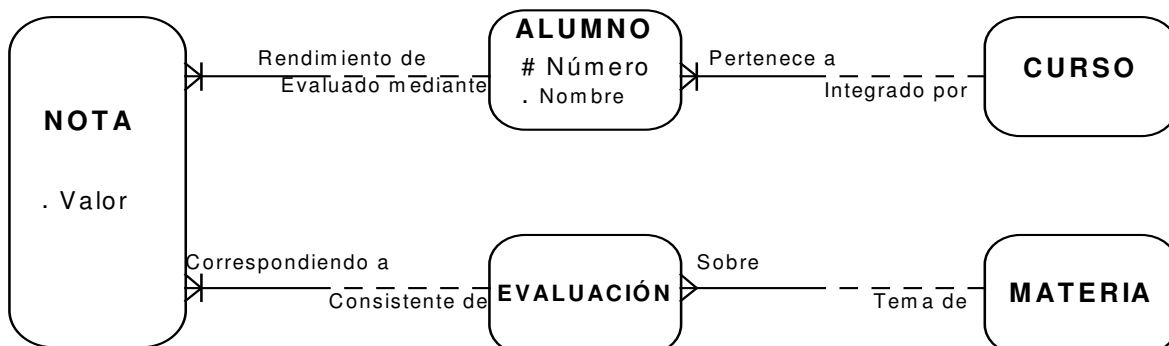
El identificador único puede ser un atributo, combinación de atributos o combinación de atributos y relaciones.

Nótese que una misma entidad puede tener más de una forma de identificación única.

Así, por ejemplo, un empleado puede identificarse tanto con su número de cédula como con un código interno asignado por la organización.

A los atributos que pueden emplearse alternativamente como identificadores de una entidad se les denomina claves candidatas.

El método básico de representar el identificador único en un diagrama entidad relación consiste en anteceder del signo # a cada atributo que contribuya al identificador único y en colocar una barra cruzada en la línea de la(s) relación(es) que participa(n) del identificador.



En este diagrama, para identificar a un alumno, se requiere el código de curso y el

número de lista, en tanto que, para identificar la nota, se requiere el identificador de alumno unido con el identificador de evaluación.

4. EXTENSIÓN AL CASO DE ESTUDIO.

En nuestro ejemplo hemos apreciado las construcciones básicas empleadas para la elaboración de un modelo entidad relación, así como los conceptos básicos del Modelaje.

En esta sección extenderemos el ejemplo para los colegios de bachillerato, examinando también algunas construcciones de agregación.

4.1 PLANTEAMIENTO.

En bachillerato, un mismo profesor puede dictar una o más materias en varios cursos o paralelos, dándose también que en distintos cursos se estudien distintas materias.

Entre las entidades más relevantes de este dominio de aplicación tenemos:

CURSO	Cada uno de los grados anuales de estudio.
PARALELO	Cada uno de los grupos correspondientes a cada curso, identificados generalmente con las letras A, B, C, etc.
AULA	Espacio físico donde se aloja cada paralelo.
PROFESOR	Instructor encargado de dictar las materias correspondientes a cada curso en cada paralelo.
MATERIA	Asignatura objeto de estudio en los distintos cursos.
ALUMNO	O estudiante, asignado a cada paralelo.
EVALUACIÓN	Examen en el que se revisa el rendimiento de cada alumno en cada materia.

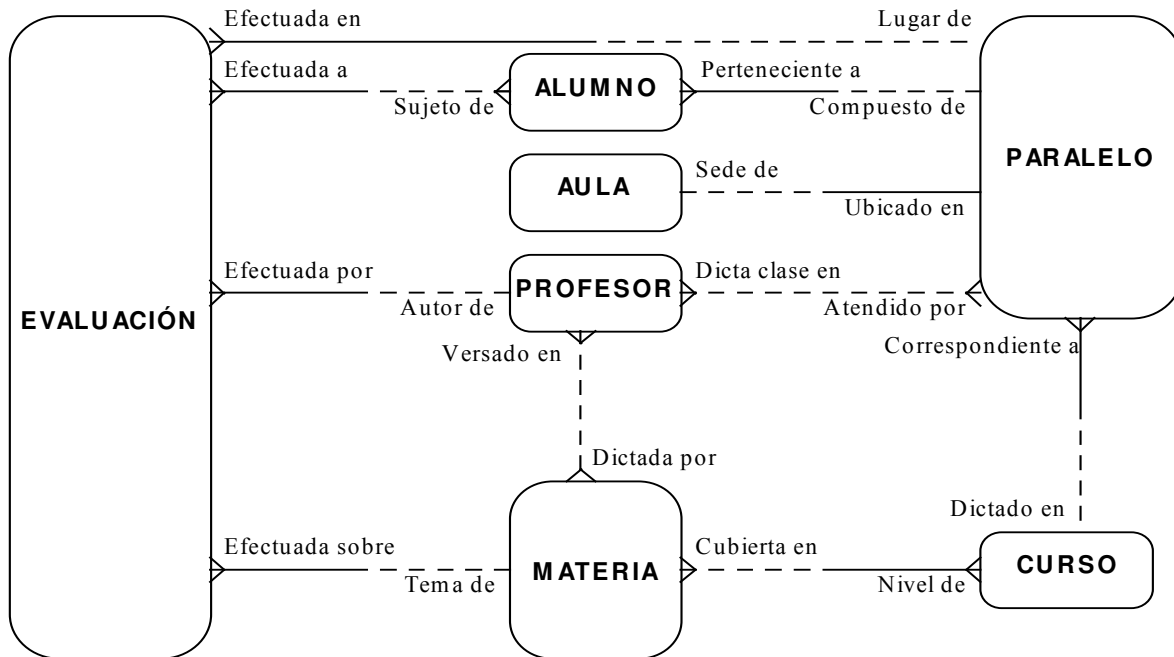
Entre estas entidades podríamos identificar, a priori, las siguientes relaciones:

CURSO/PARALELO	1:N Un mismo curso puede tener varios paralelos (nombrados A, B, C, etc.), en tanto que un paralelo debe corresponder a uno y sólo un curso.
PARALELO/AULA	1:1 Un mismo paralelo debe estar situado en una y sólo un aula, en tanto que un aula puede estar asignada a uno y sólo un paralelo.
PARALELO/ALUMNO	1:N Un paralelo debe tener uno o más alumnos, en tanto que un alumno debe pertenecer a uno y sólo un paralelo.
PROFESOR/PARALELO	1:1 Un profesor puede ser director de uno y sólo un paralelo, en tanto que un paralelo debe tener uno y sólo un director de grupo.
PROFESOR/PARALELO	M:N Un profesor puede dictar clase en uno o más

	paralelos, en tanto que en un paralelo pueden dictar clase uno o más profesores.
PROFESOR/MATERIA	M:N Un profesor puede dictar una o más materias, en tanto que una misma materia puede ser dictada por uno o más profesores.
CURSO/MATERIA	M:N En un curso se debe dictar una o más materias, en tanto una misma materia puede ser dictada en uno o más cursos.
MATERIA/EVALUACIÓN	1:N Una misma materia puede ser objeto de una o más evaluaciones, en tanto una evaluación debe corresponder a una y sólo una materia.
PARALELO/EVALUACIÓN	1:N En un paralelo se pueden efectuar una o más evaluaciones, en tanto que una evaluación debe efectuarse en uno y sólo un paralelo.
PROFESOR/EVALUACIÓN	1:N Un profesor puede efectuar una o más evaluaciones, en tanto que cada evaluación debe haber sido efectuada por uno y sólo un profesor.
ALUMNO/EVALUACIÓN	M:N Un alumno puede presentar una o más evaluaciones, en tanto que una evaluación debe llevarse a cabo para uno o más alumnos.

Atención! Las relaciones mencionadas constituyen únicamente un punto de partida en el análisis y resultan de nuestra comprensión intuitiva del dominio de aplicación, siendo necesario refinarlas para llegar a un modelo completo y consistente.

El diagrama entidad/relación correspondiente a las entidades y relaciones identificadas es:



Examinando algunas de estas relaciones encontramos casos especiales dignos de discusión.

4.2 DIFERENTES RELACIONES DEFINIDAS SOBRE LAS MISMAS ENTIDADES.

Nótese que entre las entidades PROFESOR y PARALELO existen dos relaciones de diferente significado:

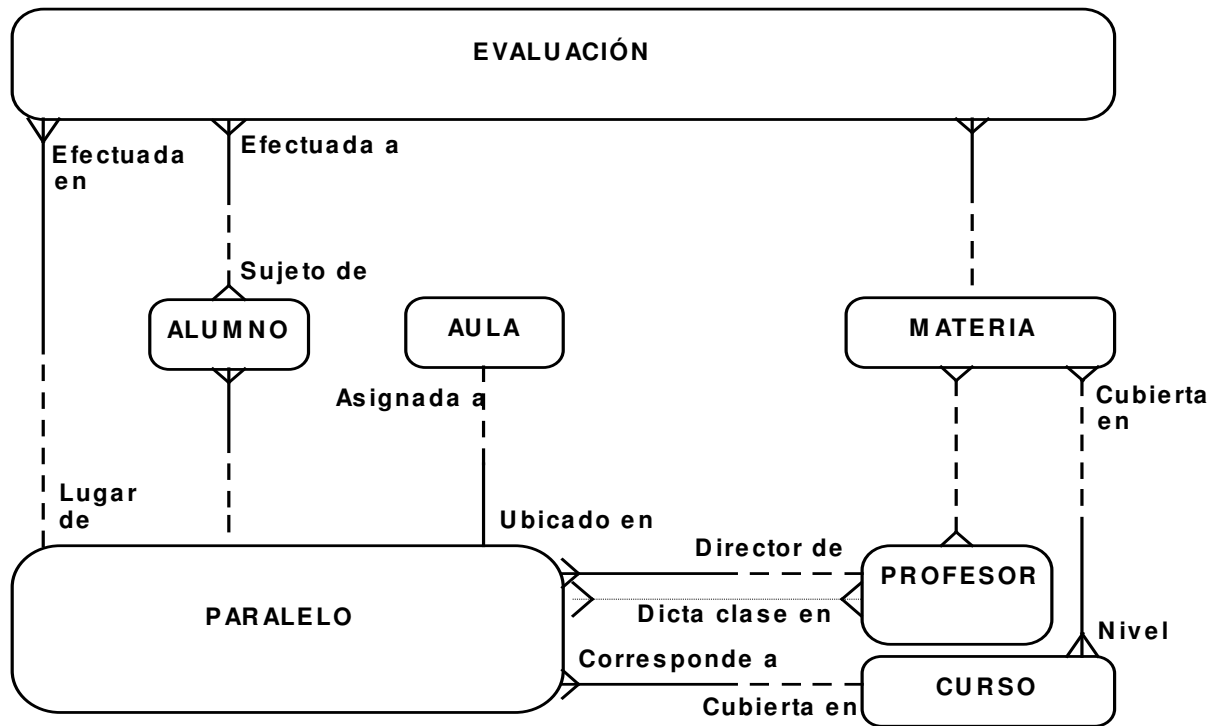
- Ser director de Grupo.
- Dictar clase en.

No es necesario dictar clase en un paralelo para dirigirlo, de la misma forma que es posible dictar clase en un paralelo y no ser director del mismo.

La existencia de varias relaciones diferentes definidas sobre las mismas entidades es bastante común y no se debe desestimar su posible ocurrencia en un modelo entidad/relación.

4.3 REEMPLAZO DE RELACIONES 1:1 POR RELACIONES 1:N.

En algunos casos, puede presentarse que la relación ser director de sea 1:N; esto es, que un mismo profesor pueda ser (simultáneamente) director de varios paralelos. En tal caso, la forma 1:N siempre podrá expresar también una relación 1:1, pudiendo, por tanto, sustituirla.

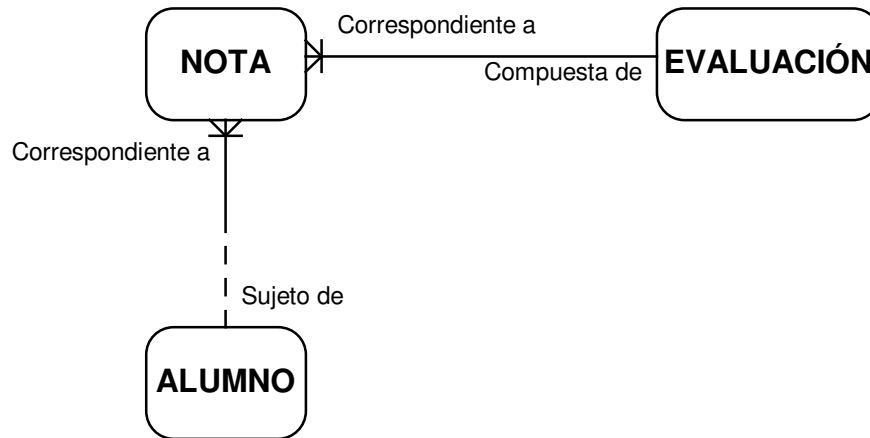


4.4 ENTIDADES DE INTERSECCIÓN.

En la variedad de Modelaje entidad/relación estudiada en este documento se trata a las relaciones M:N como entidades de intersección; esto es, para toda relación M:N identificada en el modelo se crea una nueva entidad que "intersecta" a las entidades participantes en la relación M:N y cuyo identificador único (clave primaria) se forma mediante la combinación de las claves primarias de dichas entidades.

La operación de considerar una relación M:N como entidad en sí misma se conoce como agregación.

Aplicando este procedimiento de resolución a la relación M:N existente entre ALUMNO y EVALUACIÓN, se crea una nueva entidad (NOTA) que intersecta a las dos anteriores, como se aprecia en el diagrama.



Examinando las relaciones:

CURSO/MATERIA Pensum.

PROFESOR/MATERIA Estar capacitado para dictar.

PROFESOR/PARALELO Dictar clase en.

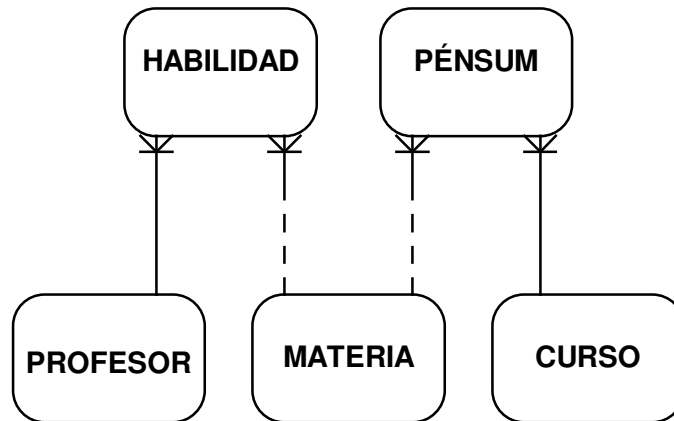
encontramos que NO son relaciones independientes; esto es, no cualquier profesor puede dictar cualquier materia en cualquier paralelo.

De hecho, un profesor sólo puede dictar, en un paralelo dado, las materias que este capacitado para dictar en cada curso.

Así mismo, en un paralelo dado, únicamente se pueden estudiar aquellas materias definidas para el curso correspondiente al paralelo.

Esta dependencia corresponde a una forma de relación conocida como relación ternaria; esto es, una relación definida simultáneamente sobre tres entidades y que no siempre se puede descomponer en tres relaciones binarias independientes sin pérdida de información.

Aplicando la técnica de resolución a las relaciones M:N CURSO/MATERIA y PROFESOR/MATERIA, se define las entidades de intersección PENSUM y HABILIDAD, respectivamente.



Nótese en esta figura el uso de las barras perpendiculares trazadas sobre las líneas de relación.

Una barra perpendicular a la línea de relación indica que la entidad lado N construye su clave primaria involucrando la clave primaria de la entidad lado 1.

Como se aprecia en el diagrama, este es siempre el caso con las entidades de intersección.

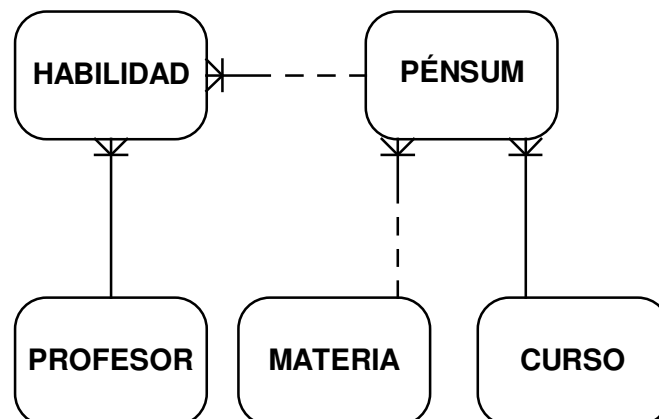
Este tipo de relación se conoce también como relación de dependencia y a la entidad de intersección se le denomina débil respecto a la(s) entidad(es) lado 1.

El modelo elaborado hasta ahora, sin embargo, contiene aún un defecto fundamental.

Un profesor habilitado para dictar matemáticas en 3o (álgebra) no necesariamente está habilitado para dictarlas en 6o (cálculo).

Es decir, no existe una relación directa entre PROFESOR y MATERIA, sino entre PROFESOR y PÉNSUM (esto es, simultáneamente con MATERIA y CURSO).

Reemplazando la relación entre HABILIDAD y MATERIA por la relación HABILIDAD y PÉNSUM, tenemos el diagrama siguiente:



En este diagrama reflejamos el hecho de que un PROFESOR puede estar habilitado para dictar la MATERIA matemáticas en los CURSOS 3o y 4o (álgebra), pero no en los CURSOS 5o y 6o (trigonometría y cálculo).

De esta manera, un profesor no se relaciona de forma directa con la entidad MATERIA en general, sino con la intersección PENSUM existente entre MATERIA y CURSO.

Esta técnica permite resolver relaciones ternarias en relaciones binarias sin pérdida de información.\

La construcción de "promover" una relación M:N a una entidad y definir nuevas relaciones sobre dicha entidad se denomina agregación y representa una de las herramientas conceptuales más poderosas del Modelaje entidad/relación.

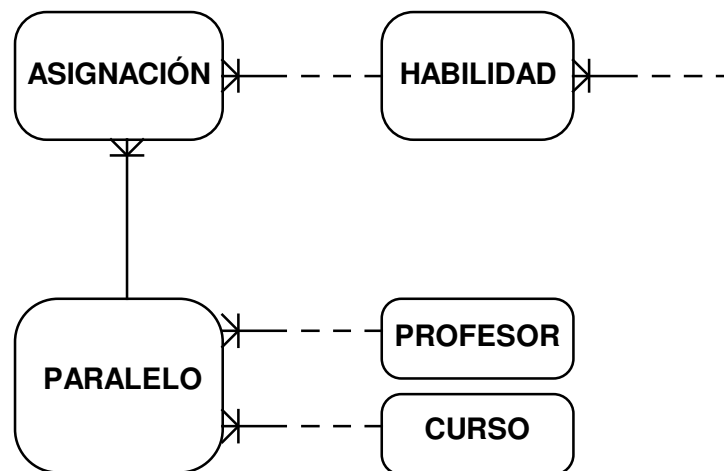
Aplicando mecánicamente esta transformación a la relación M:N existente entre PROFESOR y PARALELO (dictar clase en), se podría definir una entidad de intersección, denominada ASIGNACIÓN, que resolvería la relación.

Esta entidad de intersección, sin embargo, no se puede definir directamente entre PROFESOR y PARALELO, pues permitiría asociar cualquier PROFESOR con cualquier PARALELO.

Un análisis mas cuidadoso revela que sólo es posible asignar un PROFESOR a un PARALELO si el PROFESOR está habilitado par dictar la MATERIA dada en el CURSO correspondiente al PARALELO.

Debido a ello, ASIGNACIÓN debe intersectar las entidades PARALELO y HABILIDAD, en vez de PARALELO y PROFESOR.

Es de notar que (aunque el modelo no lo determina explícitamente), el CURSO correspondiente al PENSUM asociado con la HABILIDAD del PROFESOR debe ser el mismo CURSO correspondiente al PARALELO con el cual se asocia la HABILIDAD.



Esta construcción se denomina clausura transitiva de la relación.

Por último, las relaciones directas

MATERIA/EVALUACIÓN Ser tema de

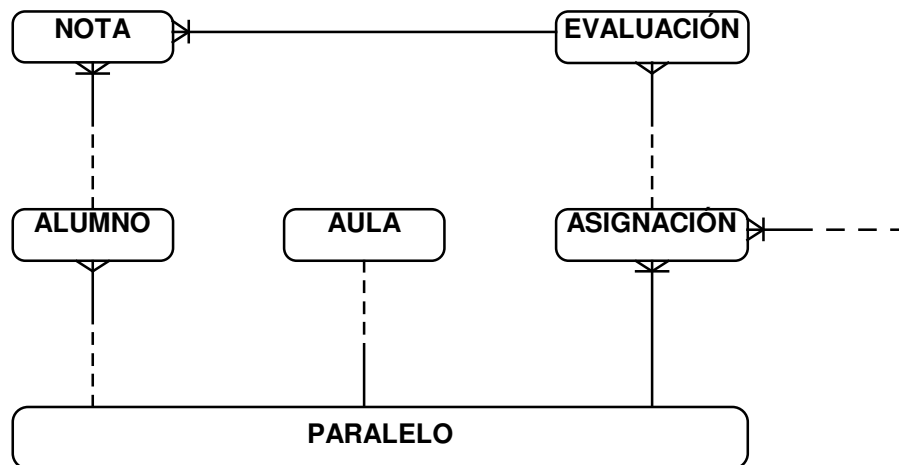
PROFESOR/EVALUACIÓN Ser autor de

PARALELO/EVALUACIÓN Ser lugar de

No son tampoco independientes entre sí.

De hecho, sólo el PROFESOR que haya sido asignado para dictar MATERIAS en un PARALELO particular (mediante la entidad de intersección ASIGNACIÓN) puede originar EVALUACIONES.

Estas relaciones, por tanto, pueden reemplazarse por una única relación entre ASIGNACIÓN y EVALUACIÓN como se ilustra en el diagrama.



Es de notar que el paralelo correspondiente a la EVALUACIÓN (a través de la entidad ASIGNACIÓN) debe corresponder al PARALELO de los ALUMNOS participantes en la EVALUACIÓN.

Este es, nuevamente, el caso de una clausura transitiva de la relación.

4.5 MODELO FINAL.

Después de estas transformaciones, resultantes de un análisis más detallado, tenemos como lista de entidades:

CURSO Cada uno de los grados anuales de estudio.

PARALELO Cada uno de los grupos correspondientes a cada curso, identificados generalmente con las letras A, B, C, etc..

AULA Espacio físico donde se aloja cada paralelo.

PROFESOR Instructor encargado de dictar las materias correspondientes a cada curso en cada paralelo.

MATERIA	Asignatura objeto de estudio en los distintos cursos.
ALUMNO	O estudiante, asignado a cada paralelo.
PÉNSUM	Materias cubiertas por el programa de un curso particular.
HABILIDAD	Materias susceptibles de ser dictadas por un profesor en un curso particular.
ASIGNACIÓN	Asociación entre un profesor, una materia en un curso particular y un paralelo dado.
EVALUACIÓN	Examen en el que se revisa el rendimiento de cada alumno en cada materia.
NOTA	Rendimiento obtenido por un alumno dado en una evaluación particular.

La lista de relaciones identificadas es:

CURSO/PARALELO	1:N Un mismo curso puede tener varios paralelos (nombrados A, B, C, etc.), en tanto que un paralelo debe corresponder a uno y sólo un curso.
PARALELO/AULA	1:1 Un mismo paralelo debe estar situado en una y sólo un aula, en tanto que un aula puede estar asignada a uno y sólo un paralelo.
PARALELO/ALUMNO	1:N Un paralelo puede tener uno o más alumnos, en tanto que un alumno debe pertenecer a uno y sólo un paralelo.
CURSO/PENSUM	1:N con dependencia. Un curso debe estar presente en varias líneas de pensum, en tanto que cada línea de pensum debe referenciar a uno y sólo un curso.
MATERIA/PENSUM	1:N con dependencia. Una materia puede estar presente en varias líneas de pensum, en tanto que cada línea de pensum debe referenciar a una y sólo una materia.
PENSUM/HABILIDAD	1:N con dependencia. Una línea de pensum (curso, materia) puede estar presente en varias líneas de habilidad, en tanto que una línea de habilidad debe corresponder a una y sólo una línea de pensum.
PROFESOR/HABILIDAD	1:N con dependencia. Un profesor debe estar asociado a varias líneas de habilidad, en tanto que una línea de habilidad debe estar asociada con uno y sólo un profesor.
HABILIDAD/ASIGNACIÓN	1:N con dependencia. Una línea de habilidad (profesor, materia, curso) puede estar asociada con varias asignaciones, en tanto que cada línea de asignación debe estar asociada con una y sólo una habilidad.

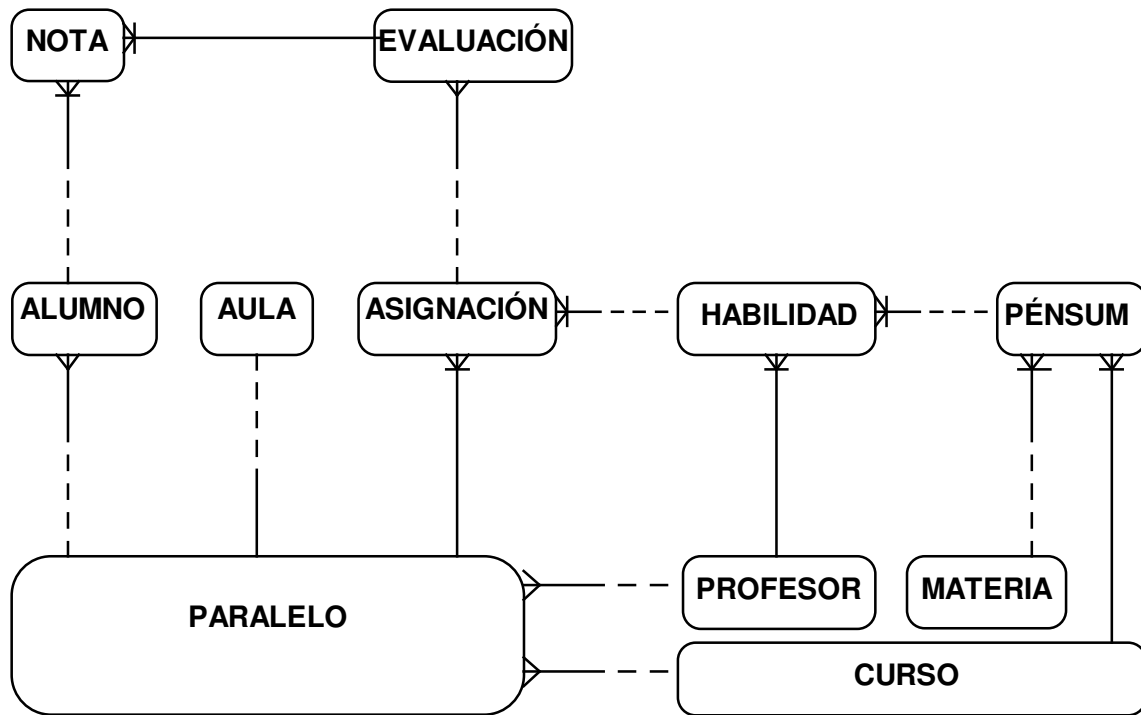
PARALELO/ASIGNACIÓN 1:N con dependencia. Un paralelo debe estar asociado con una o más líneas de asignación, en tanto que una línea de asignación debe referenciar a uno y sólo un paralelo.

ASIGNACIÓN/EVALUACIÓN 1:N Una misma asignación (profesor, materia, curso, paralelo) puede ser objeto de una o más evaluaciones, en tanto que una evaluación debe corresponder a una y sólo una asignación.

EVALUACIÓN/NOTA 1:N con dependencia. Una misma evaluación debe tener una o más notas, en tanto que una nota debe corresponder a una y sólo una evaluación.

ALUMNO/NOTA 1:N con dependencia. un alumno puede tener una o más notas, en tanto que una nota debe corresponder a uno y sólo un alumno.

El diagrama completo para el modelo en estudio es:



5. IDENTIFICACIÓN DE ENTIDADES, RELACIONES Y ATRIBUTOS.

5.1 INTRODUCCIÓN.

En el ejemplo anterior, hemos efectuado un análisis de entidades y relaciones, sin abordar aún la identificación de atributos.

Como puede apreciarse, el Modelaje entidad/relación no es una técnica de aplicación mecánica que provea, mediante una secuencia estricta y simple de actividades, el modelo preciso.

Excepto en los casos más simples, siempre resulta necesario revisar varias veces el modelo resultante del análisis para capturar con la mayor exactitud posible las propiedades del dominio modelado.

Esta tarea involucra muy de cerca al usuario final, como único árbitro definitivo de las dudas e inconsistencias encontradas, así como para la eliminación de relaciones irrelevantes.

Así, en el ejemplo anterior, no se contemplaron relaciones carentes de importancia para el dominio de aplicación tales como:

ALUMNO/MATERIA Qué relación existe?. Quizás que el alumno dado es muy aficionado a la materia. Es esta relación relevante?. Podría serlo si el sistema incluyera datos de tipo vocacional, pero no si se limita a calificaciones.

PROFESOR/ALUMNO Qué relación existe?. Quizás que el alumno y el profesor son buenos amigos. Es esta relación relevante?.

La inclusión de relaciones redundantes es uno de los errores más comunes en el Modelaje.

Este es el caso con las dos relaciones anteriormente citadas. Un analista principiante podría sentirse tentado a incluir estas relaciones para responder a preguntas tales como:

- Cuál es la nota promedio de un ALUMNO dado para una MATERIA dada?.
- Qué profesores dictan clase sobre cada MATERIA a qué ALUMNOS?.

Nótese, sin embargo, que para responder estas preguntas basta con recorrer las relaciones ya definidas en el orden apropiado, siendo por tanto, redundante establecer relaciones directas entre las entidades involucradas.

Así, dado un ALUMNO, se puede enlazar todas sus NOTAS, seleccionando para cada una únicamente aquellas correspondientes a una MATERIA dada y así calcular la nota promedio.

Igualmente, dado un PROFESOR, se puede enlazar todas sus HABILIDADES, enlazando a continuación sus ASIGNACIONES para determinar el PARALELO, a partir del cual se pueden enlazar todos los ALUMNOS.

Podría alegarse, quizás, que las primeras relaciones citadas en la sección anterior (aquellas identificadas a priori) eran suficientes para construir un sistema automatizado, sin entrar en las sutilezas de qué profesor puede dictar qué materia o qué materias se

cubren en qué cursos.

Lo esencial en un sistema automático, sin embargo, no es sólo la habilidad de almacenar los datos propios del dominio de aplicación , sino también (y muy especialmente) garantizar su consistencia.

A este respecto podría decirse que uno de los objetivos centrales del Modelaje es modelar en la estructura de datos (NO en los programas de aplicación) la mayor cantidad posible de propiedades dinámicas del dominio de aplicación.

Así, se garantiza que el modelo de datos hace imposible expresar inconsistencias que, de otra manera, deberían ser chequeadas por los programas de aplicación.

Esto, por supuesto, redundará en una reducción de la complejidad y dificultad de mantenimiento de los programas.

Extendiendo este concepto, es fácil observar que una gran cantidad de lógica procedimental necesaria en los programas de aplicación puede obviarse mediante la simple inclusión de entidades y relaciones de carácter abstracto en el modelo de datos. (Véase la sección 6.2. Entidades de Referencia).

5.2 PASOS DEL MODELAJE.

Una vez dotados de la comprensión básica de los conceptos y construcciones del Modelaje entidad/relación podemos planear la secuencia de pasos necesaria para elaborar un modelo completo.

Estos pasos son:

1. Identificación entidades.
2. Identificación relaciones.
3. Elaborar el diagrama entidad/relación.
4. Identificar atributos.
5. Refinar el modelo, comparándolo contra los requerimientos establecidos para el dominio de aplicación.
6. Transformar el modelo entidad/relación a un diseño lógico implementable. Por ejemplo, a un esquema de tablas, vistas e índices de una base de datos relacional.

5.2.1 Identificación de entidades.

La identificación de entidades es el paso central del proceso de Modelaje. Una simplificación útil de los diversos tipos de entidad es:

Personas naturales o jurídicas, tales como ALUMNO, PROFESOR, CLIENTE, EMPLEADO, PROVEEDOR, etc..

Objetos Tangibles o no, tales como ARTÍCULO, CUENTA CORRIENTE, CUENTA CONTABLE, etc..

Transacciones	O eventos, tales como EVALUACIÓN, COMPRA, DESPACHO, CONSIGNACIÓN, TRANSFERENCIA, etc..
Lugares	Tales como AULA, CIUDAD, BODEGA, etc..
Conceptos	O abstracciones, tales como TIPO DE EVALUACIÓN, TIPO DE CLIENTE, TIPO DE TRANSACCIÓN, UNIDAD DE MEDIDA, etc..

En términos generales, se presenta que sólo las personas pueden originar transacciones y que la acción de estas recae siempre sobre un objeto dado.

Por otra parte, las personas (y objetos) suelen residir (o almacenarse) en lugares.

Así mismo, los conceptos permiten controlar y limitar qué tipo de persona puede llevar a cabo qué tipo de transacción y qué tipo de transacción se puede aplicar sobre cada tipo de objeto.

5.2.2 Identificación de relaciones.

En un sistema comercial típico (transaccional) una buena guía para iniciar al análisis es preguntarse:

- Qué transacción ocurre en el modelo?. Transacciones: VENTA, DESPACHO, TRANSFERENCIA, ASIENTO CONTABLE, etc..
- Quién origina cada transacción?. Personas: CLIENTE, PROFESOR, PROVEEDOR, etc..
- Sobre qué objeto recae la acción de la transacción?. Objetos: ARTÍCULO, CUENTA CORRIENTE, CUENTA CONTABLE, etc..
- De qué lugares provienen las personas, en qué lugares se almacenan o procesan los objetos?. Lugares: BODEGA para ARTÍCULO, CIUDAD para PROVEEDOR, AULA para PARALELO, etc..
- De qué depende la naturaleza de las transacciones, personas u objetos?. Conceptos: UNIDAD para ARTÍCULO, TIPO DE TRANSACCIÓN para TRANSACCIÓN BANCARIA, CLASE DE CLIENTE para CLIENTE, etc..

Véase la sección 7. (Estructuras Clásicas y Patrones Genéricos) para obtener información acerca de construcciones recurrentes, cuyo conocimiento simplifica el análisis en un nuevo dominio de aplicación.

5.2.3 Elaboración del diagrama entidad/relación.

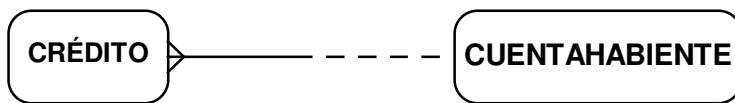
La importancia de los diagramas entidad/relación no debe ser, de ninguna manera, desestimada, como (desafortunadamente) ocurre con la documentación en los sistemas convencionales.

El diagrama entidad/relación no es únicamente un medio de documentación o revisión rápida. Es, ante todo, un instrumento de comunicación con el usuario.

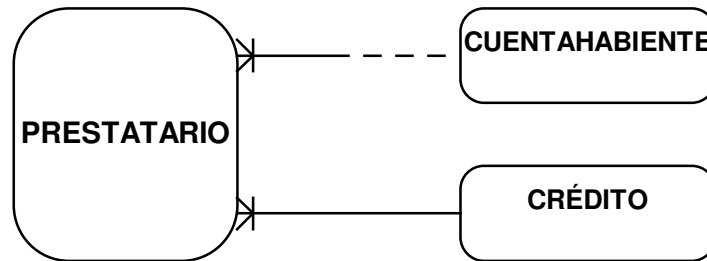
Una experiencia de años en el Modelaje entidad/relación muestra que los usuarios finales aprenden con mucha facilidad a interpretar diagramas entidad/relación, validando la completitud y consistencia del mismo, dirimiendo diferencias e identificando formas más adecuadas de modelar.

Veamos algunos ejemplos de observaciones hechas por usuarios a modelos entidad/relación sometidos a su revisión:

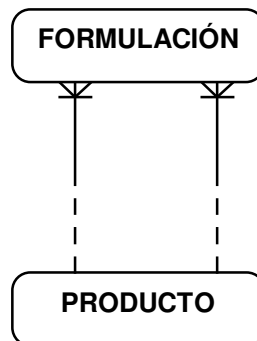
1. "Si, entiendo correctamente lo que dice el diagrama, creo que es un error considerar que un crédito pueda ser conferido a uno y sólo un cuentahabiente. Es perfectamente posible que dos o más clientes del banco apliquen para un mismo crédito, en cuyo caso se hacen igualmente solidarios por la deuda".



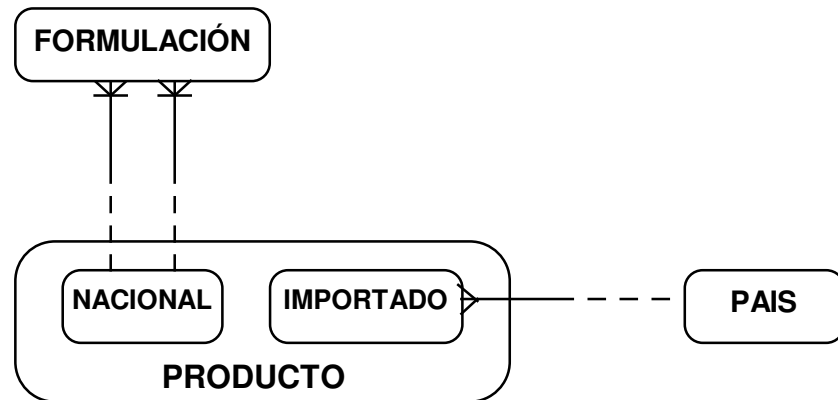
Una relación inicialmente identificada como 1:N resulta ser M:N y requiere, por tanto, de una nueva entidad (de intersección).



2. "No sé si me equivoco, pero no creo que todos los productos tengan una formulación. Nosotros vendemos también compuestos importados para los cuales ni conocemos ni nos interesa la formulación".



Una relación debe resultar una relación puede y se introduce una nueva entidad (producto importado) con atributos y relaciones propias.



Valide siempre sus diagramas entidad/relación con el usuario final !.

5.2.4 Identificación de atributos.

La identificación de atributos es el paso siguiente después de la elaboración (y aprobación) del diagrama entidad/relación.

Es posible que durante este proceso se identifiquen nuevas entidades y relaciones, sutilmente ocultas en el modelo original.

No se desanime. Esa es la naturaleza progresiva del Modelaje entidad/relación.

5.2.4.1 Identificador único.

El primero de los atributos que debe ser identificado para cada entidad es el identificador o clave primaria.

Toda entidad debe tener una clave primaria. En los raros casos en que el Modelaje no evidencie un atributo (o combinación de atributos) que conformen un identificador único, es lícito "inventar" uno utilizando, por ejemplo, números consecutivos, como en los documentos.

5.2.4.2 Número de atributos.

Un chequeo de calidad importante es verificar que toda entidad tenga, por lo menos, dos atributos: La clave primaria y un descriptor.

Este criterio no se aplica necesariamente a las entidades de intersección, que pueden consistir, únicamente, de las claves primarias de las entidades intersectadas.

5.2.4.3 Caso de estudio.

La identificación de atributos par nuestro caso de estudio sería:

CURSO
número
nombre

PARALELO

letra

AULA

número

capacidad (en número de alumnos)

PROFESOR

número

nombre

sexo

ALUMNO

número

nombre

sexo

MATERIA

código

nombre

área (sociales, técnicas, humanidades, etc.)

PENSUM

Intensidad (en número de horas por semana)

HABILIDAD

Calificación (excelente, bueno, aceptable)

ASIGNACIÓN

fecha_inicial

fecha_final

EVALUACIÓN

número

tipo de evaluación (mensual, semestral, final)

fecha

NOTA

calificación (por ejemplo, de 1 a 10)

Conceptos avanzados.

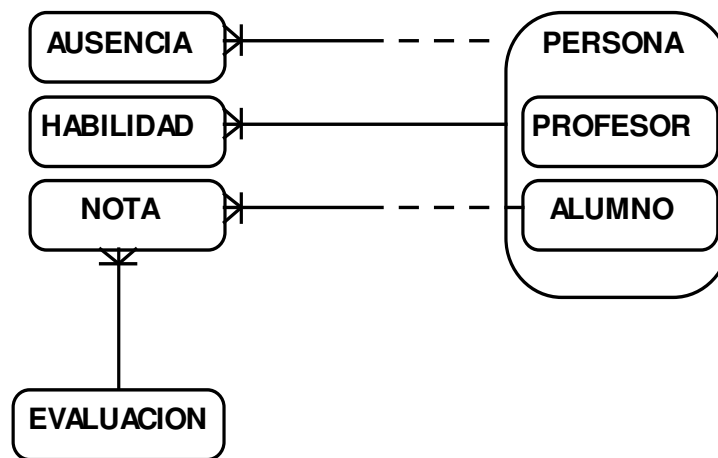
5.3 SUPERTIPOS.

En la lista anterior, podemos apreciar que los atributos de las entidades PROFESOR y ALUMNO son los mismos, lo que se deriva del hecho natural de que ambos son personas, resultando razonable agrupar estas dos entidades en una nueva entidad.

Cuando dos o más entidades comparten atributos (o relaciones) comunes se configura un supertipo.

Un subtipo es un tipo de entidad; esto es, un componente de un supertipo.

En la siguiente figura se presenta el supertipo PERSONA, compuesto de los subtipos ALUMNO y PROFESOR.



El estándar diagramático para la representación de supertipos es presentar las cajas correspondientes a las entidades subtipo dentro de la caja correspondiente a la entidad del supertipo.

Así, una entidad puede dividirse en dos o más subtipos mutuamente exclusivos, cada una de las cuales posee atributos o relaciones comunes con las demás.

Estos atributos o relaciones se definen explícitamente sólo al nivel superior del supertipo.

Por supuesto, los subtipos pueden tener atributos o relaciones propias.

Así mismo, un subtipo puede ser dividido en otros subtipos y así sucesivamente. La experiencia muestra, sin embargo, que dos o tres niveles son suficientes en la mayoría de las circunstancias.

Si un subtipo no resulta tener atributos o relaciones propias, puede tratarse de un sinónimo o un rol jugado por la entidad, pero que no lo diferencia como entidad separada.

Es posible que dos o más subtipos de un mismo supertipo posean relaciones en sí o con el supertipo.

Siempre que se defina un supertipo, es necesario añadirle un atributo (o tipificador) que indique a qué subtipo particular pertenece una instancia dada del supertipo.

Este atributo es siempre requerido; es decir, no se lo puede considerar opcional.

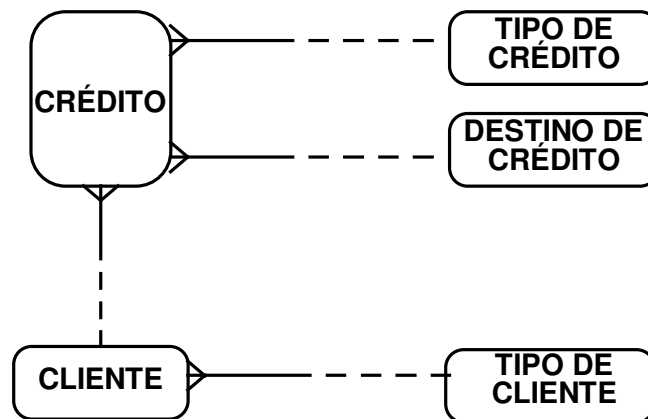
Ocasionalmente, es conveniente tener subtipos superpuestos u ortogonales, en los que una misma instancia del supertipo puede pertenecer, simultáneamente, a varios subtipos, como el supertipo PERSONA que puede ser simultáneamente EMPLEADO, CLIENTE y PROVEEDOR, en aquellos casos en que un empleado de la compañía puede vender insumos a la misma o un cliente que suele comprar determinados productos venda también a la compañía otros.

En este caso, se suele emplear distintos colores para las cajas de cada uno de los subtipos.

5.4 ENTIDADES DE REFERENCIA.

Una entidad de referencia es aquella que no tiene extremos de relaciones 1:N conectados a ella, esto es, una entidad que es siempre lado 1 de alguna relación y lado N de ninguna. Generalmente, hay pocas entidades de esta naturaleza en un diagrama entidad/relación típico.

Estas entidades suelen corresponder a abstracciones o conceptos tales como TIPO DE CLIENTE (cuentahabiente, empleado, particular, etc.), TIPO DE CRÉDITO (hipotecario, prendario, quirografario, etc.) y DESTINO DE CRÉDITO (consumo, inversión, fomento, etc.) como en el diagrama de la figura.



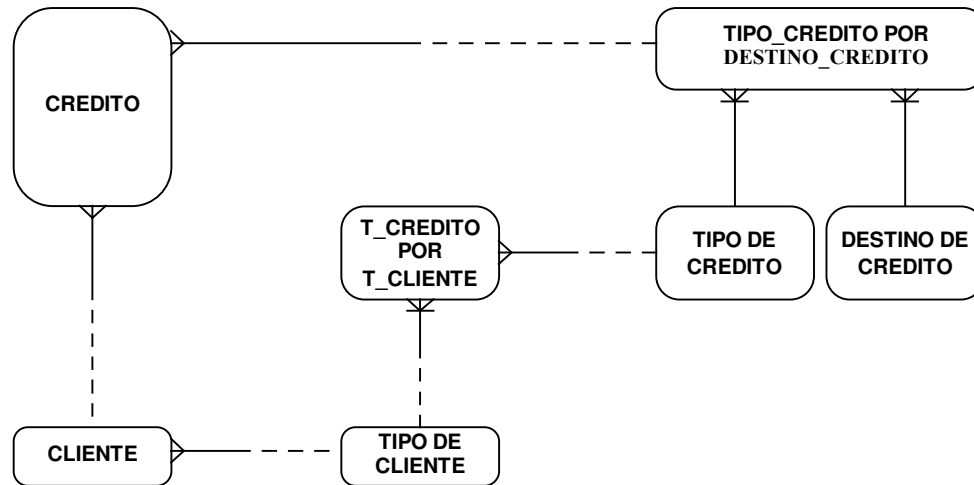
Nótese que estas entidades abstractas suelen tener como únicos atributos un identificador y una descripción, sirviendo tan sólo para cualificar a otra entidad.

Su importancia, sin embargo, no debe ser desestimada pues el Modelaje de relaciones entre ellas permite, en la mayoría de los casos, expresar en el modelo de datos propiedades que requerirían extensa lógica procedimental en los programas de aplicación.

Así, por ejemplo, puede establecerse que no cualquier TIPO DE CLIENTE puede aplicar

para TIPO DE CRÉDITO y que no todo TIPO DE CRÉDITO es aplicable para todo DESTINO.

Dadas estas consideraciones, el diagrama entidad/relación podría modificarse como aparece en la figura.



Nótese que valores tales como la tasa de interés o el plazo máximo de pago pueden asociarse con las entidades de intersección identificadas, de forma que, por ejemplo, la tasa de interés dependa de la relación entre TIPO DE CLIENTE y DESTINO DE CRÉDITO y el plazo de pago de la relación entre TIPO DE CLIENTE y TIPO DE CRÉDITO.

Estas relaciones (y sus atributos) permiten definir reglas de la organización dentro del modelo de datos, logrando el efecto de:

- Liberar a los programas de aplicación de lógica compleja.
- Permitir cambios en las políticas de la compañía sin tener que modificar los programas.

En aplicaciones convencionales (escritas en lenguajes de tercera generación), estas reglas suelen "quemarse" dentro de la lógica de programas de aplicación, como se ilustra en el siguiente ejemplo:

```

/*Tipos de cliente*/
#define CUENTAHABIENTE          ('C')
#define PARTICULAR              ('P')
#define EMPLEADO                ('E')

/*Tipos de crédito*/
#define PRENDARIO               ('P')
#define HIPOTECARIO            ('H')
#define QUIROGRAFICO           ('Q')

/*Destinos de crédito*/
  
```

```

#define CONSUMO                ('C')
#define INVERSIÓN              ('Y')

.....

switch (tipo_cliente) {
case CUENTAHABIENTE:
    switch (tipo_credito) {
    case PRENDARIO:
        plazo_maximo = 120;
        break;
    case HIPOTECARIO:
        plazo_maximo = 360;
        break;

    case QUIROGRAFICO:
        plazo_maximo = 60;
        break;
    }
    break;

case PARTICULAR:
    if (tipo_credito == QUIROGRAFICO) {
        error("Tipo de crédito no aplicable");
        break;
    }
    switch (destino_credito) {
    case CONSUMO:
        tasa_interes = 40.0;
        break;

    case INVERSION:
        tasa_interes = 37.5;
        break;
    }
    break;
case EMPLEADO:
    ...

}

```

Obviamente, en este esquema, todo cambio en las políticas de asignación de crédito y en los valores de plazos máximos o tasas de interés se traducirán en cambios en los programas de aplicación.

Haciendo uso de las entidades de referencia identificadas y sus relaciones, la lógica del programa de aplicación se hace mucho más genérica y flexible como se muestra a continuación:

```

EXEC SQL SELECT          plazo_maximo
      INTO                :plazo_maximo

```

```

FROM      cliente_credito
WHERE     tipo_cliente =: tipo_cliente AND
          tipo_credito =: tipo_credito;

if (sqlca.sqlcode == NO_ENCONTRADO) {
    error("Tipos de cliente y crédito no coinciden");
    return(ERROR);
}

EXEC SQL SELECT      tasa_interes
INTO                 :tasa_interes
FROM                 cliente_destino
WHERE                tipo_cliente =: tipo_cliente AND
                    destino_credito =: destino_credito;

if (sqlca.sqlcode == NO_ENCONTRADO) {
    error("Tipos de cliente y destino crédito no coinciden");
    return(ERROR);
}

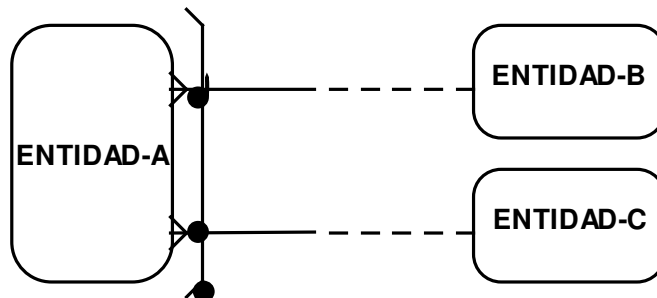
...

```

De esta manera, cualquier cambio de políticas respecto de la concesión de créditos o de valores en las tasas de interés y plazos de pago implicarán únicamente la modificación de los registros correspondientes en la base de datos y no cambios en los programas de aplicación.

5.5 RELACIONES CONDICIONALES.

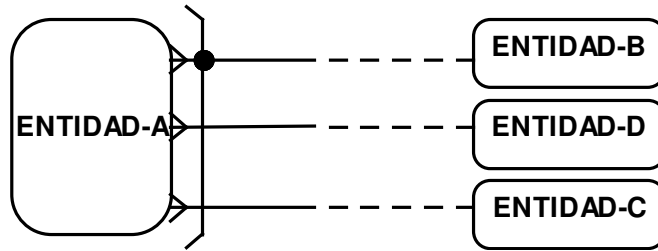
Una relación condicional es una relación en la que una instancia de la relación A puede relacionarse con una instancia de la entidad B en ciertos casos o con una instancia de la entidad C en otros, de forma excluyente, como se ilustra.



La curva trazada alrededor de las líneas de relación se denomina un arco y significa que una instancia de la entidad A puede relacionarse con una de la entidad B o con una de la entidad C, pero no con ambas a la vez.

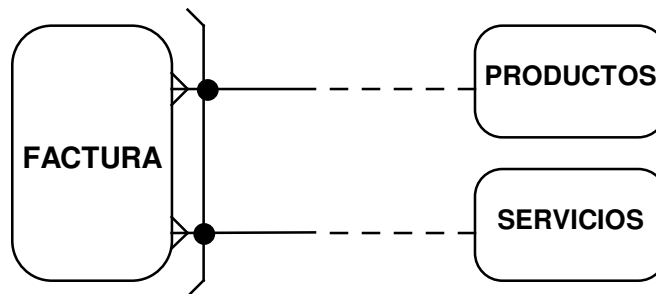
Nótese el empleo de un punto en la intersección del arco con las líneas de relación. Esta notación se emplea para evitar ambigüedades en el caso en que la entidad A esté

relacionada también con una entidad D independiente del arco, como en la figura.



En esta figura, la relación entre A y D no forma parte del arco, es decir, D no está incluida en la relación condicional entre A y B o C.

Un ejemplo de arco se tiene en una empresa de servicio automotor en la que se pueden facturar, indistintamente, productos (tales como copas, parabrisas, etc.) y servicios (tales como lavado del auto, cambio de aceite, etc.).



5.6 RELACIONES NO TRANSFERIBLES.

Normalmente, una instancia de una entidad puede conectarse a otra mediante una relación y, subsiguientemente, ser desconectada de esta y reconectada a otra instancia del mismo tipo.

Por ejemplo, un paralelo puede asociarse a un profesor como director de grupo y, posteriormente, asociarse a otro si el director original se retira del colegio.

En otros casos, sin embargo, la asociación establecida no es transferible.

Este es el caso del paralelo que está asociado con un curso. No tiene mucho sentido pensar en que, por ejemplo, a mitad de año el paralelo 4A se transforme en 5A.

Esta situación es típica en las relaciones de dependencia en las que la clave primaria del lado N se construye empleando como componente a la clave primaria del lado 1.

Nótese, en el ejemplo anterior, que el 4 de 4A corresponde al código del curso.

5.7 INTEGRIDAD REFERENCIAL.

Otra construcción asociada con las relaciones es la de integridad referencial.

La integridad referencial exige, por ejemplo, que al insertar una nueva fila en la tabla de empleados, el número de departamento de la misma exista también en la tabla de departamentos.

Qué ocurre si se borra una fila de la tabla de departamentos?.

Hay tres posibilidades para el manejo de la operación de borrado en esta circunstancia:

- Borrar a todos los empleados asociados con el departamento dado. Esta modalidad se conoce como borrado de cascada.
- Inhibir la operación de borrado del departamento si hay empleados asociados con él.
- Desconectar a los empleados asociados del departamento borrado, en espera (quizás) de que sean subsiguientemente asignados a otro departamento.

La opción de borrado de cascada se aplica cuando la relación entre el lado 1 y el lado N es debe en ambos extremos, de forma que al perder información sobre una instancia de la entidad padre se pierde también toda información sobre las instancias asociadas de la entidad hijo.

La opción de inhibir el borrado se aplica cuando la relación es puede por el lado 1 y debe por el lado N.

La opción de desconocer a las instancias del lado N se aplica cuando la relación es puede en ambos extremos.

Por lo general, la clave primaria de una entidad no cambia con el paso del tiempo y debe seleccionarse para ser lo más estable posible.

En los raros casos en que la clave primaria de una instancia de entidad cambia de valor, el cambio debe propagarse a todas las instancias de entidades asociadas con la primera a través de relaciones 1:N en las que esta es el lado 1. Esta modalidad de actualizar se conoce como actualización en cascada.

5.8 ATRIBUTOS DISCRETOS Y CONTINUOS.

Como para las entidades y relaciones, algunas propiedades de los atributos pueden emplearse para perfeccionar el modelo de datos.

Analizando los atributos identificados en el caso de estudio, podemos diferenciar dos clases principales para los mismos, dependiendo de la naturaleza de sus valores.

Los atributos continuos pueden tomar cualquier valor, generalmente dentro de un rango dado.

Ejemplo de ello son la calificación en una evaluación, que puede tomar cualquier valor comprendido entre 0 y 10 (9.8, 3.5, 7.0, etc.) y la fecha de evaluación que puede tomar cualquier valor dentro del año lectivo.

Los atributos discretos sólo pueden tomar como valor uno de los provistos en una lista predefinida de valores.

Ejemplo de ello son el sexo del alumno, que solo puede tomar los valores F (femenino) o M (masculino) y el área de la materia, que sólo puede tomar los valores S (sociales), T (técnicas) o H (humanidades).

Típicamente, los atributos continuos pueden tomar valores arbitrarios sólo dentro de un rango dado, denominado dominio, nombre que también se aplica a la lista de valores posibles de los atributos discretos.

Los dominios discretos pueden clasificarse en dos categorías: Estáticos y Dinámicos.

Un dominio discreto estático es aquel en el que el número de valores en la lista no es susceptible de variar con el tiempo, tal como sexo, que jamás tomará valores distinto de F y M.

Un dominio discreto dinámico es aquel en el que el número de valores posibles en la lista puede variar en el paso del tiempo, tal como el área de la materia, al que se podría legítimamente añadir el valor V (vocacional).

La validación de un atributo discreto estático puede almacenarse confiablemente en el diccionario de datos como parte de la definición de la tabla que lo contiene o en los programas de aplicación, pues la lógica requerida para establecer su validez no cambiará con el paso del tiempo.

Con los atributos discretos dinámicos, en cambio, se hace necesario proveer un mecanismo de verificación diferente, pues toda inserción o borrado practicados en la lista de valores válidos supondría necesariamente modificaciones en la lógica de los programas de aplicación o en el diccionario de datos.

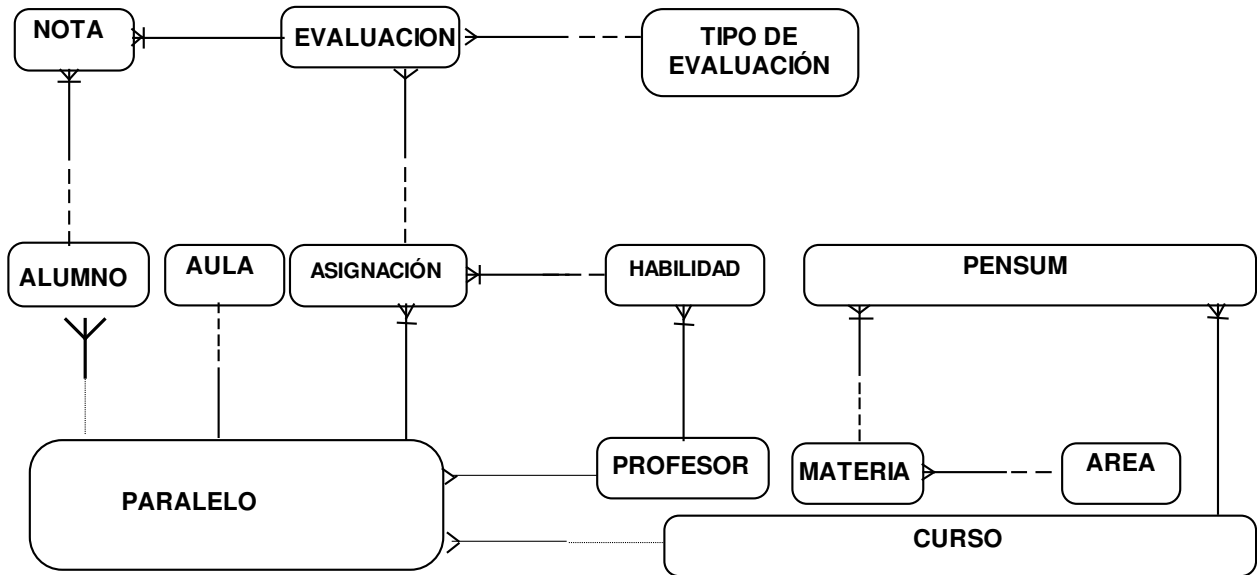
Para ganar este nivel de independencia entre datos y programas se practica la promoción de dominios discretos dinámicos a entidades de referencia.

De esta manera, lo que inicialmente se ha identificado como atributo se reemplaza por una nueva entidad (abstracta) y una relación 1:N definida entre esta y la entidad original.

Así, la lista de atributos previamente elaborada para la entidad MATERIA se transforma en:

AREA	MATERIA
# código	# código
nombre	descripción

y el diagrama entidad/relación se transforma en el mostrado a continuación.



Nótese que, en este diagrama, hemos promovido también el atributo tipo de evaluación a una entidad.

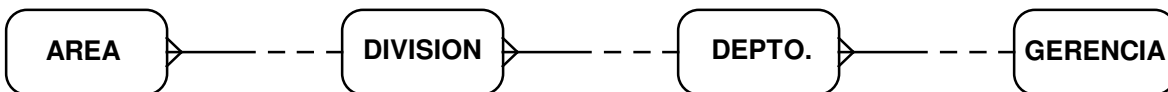
6. ESTRUCTURAS CLÁSICAS Y PATRONES GENÉRICOS.

6.1 JERARQUÍAS.

Con alguna frecuencia, se encuentran relaciones que configuran jerarquías de contenedencia tanto entre distintas entidades (como en un organigrama empresarial) como sobre la misma entidad (como en plan de cuentas contable).

El modelaje de estas jerarquías puede resultar complejo si hay un alto número de entidades involucradas en ella, como en el caso de una gran corporación con muchas subdivisiones.

Así, podemos encontrar empresas compuestas de gerencias, departamentos, divisiones y áreas, como se ilustra en el diagrama.



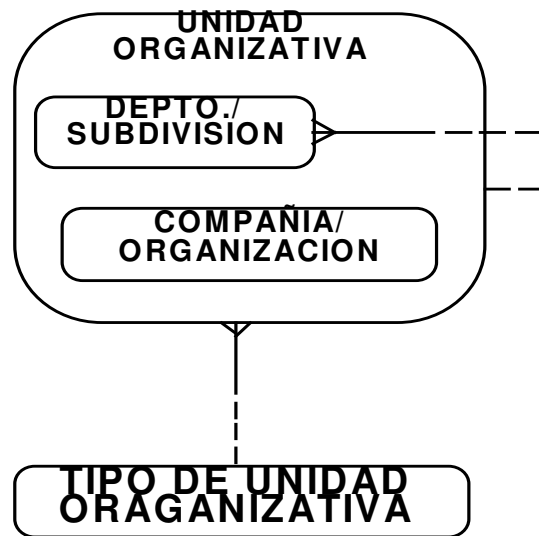
Esto puede resultar manejable, hasta que se añaden uno o más niveles.

Este diagrama puede ser modificado para admitir un número arbitrario de niveles, como se aprecia en la siguiente figura.



Este modelo es útil, pero no diferencia entre la cúspide de la jerarquía y cada uno de los niveles subordinados.

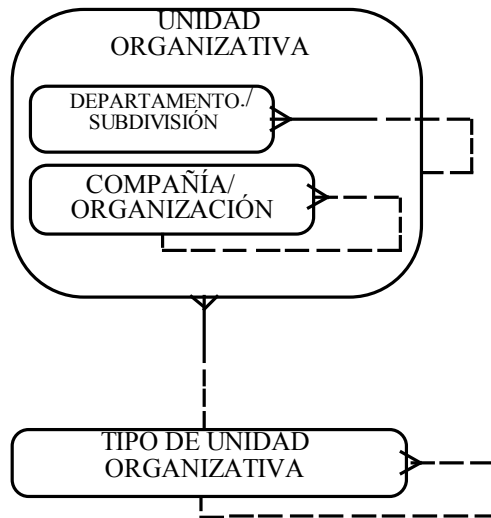
Una alternativa puede ser la construcción de un supertipo para UNIDAD ORGANIZATIVA y la introducción de una entidad de referencia para el tipo de unidad como se ilustra en el diagrama de la siguiente figura.



En este diagrama se puede manejar múltiples niveles de jerarquía y, sin embargo, diferenciar el nivel superior para modelar relaciones que sólo se aplican a él.

Sin embargo, particularmente en las aplicaciones financieras y en holdings de compañías, puede ser necesario diferenciar compañías que pertenecen a otras compañías.

En este caso se puede modelar la jerarquía de acuerdo con este diagrama.



Esta extensión reconoce que puede haber compañías dentro de la estructura de otras compañías.

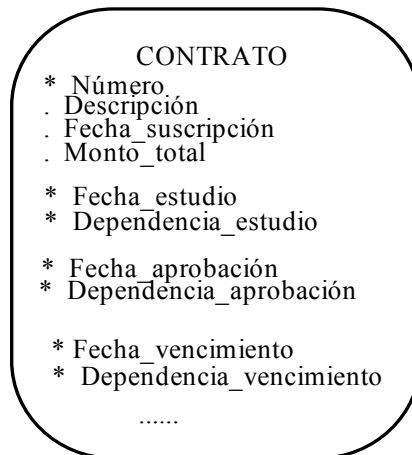
Así mismo, la relación recursiva definida sobre TIPO DE ORGANIZACION permite definir las reglas de qué clase de unidad puede existir debajo de qué otra para cada tipo de compañía.

6.2 TRANSICIÓN DE ESTADOS

En ocasiones, es preciso modelar las relaciones entre instancias de dos o más entidades a lo largo del tiempo.

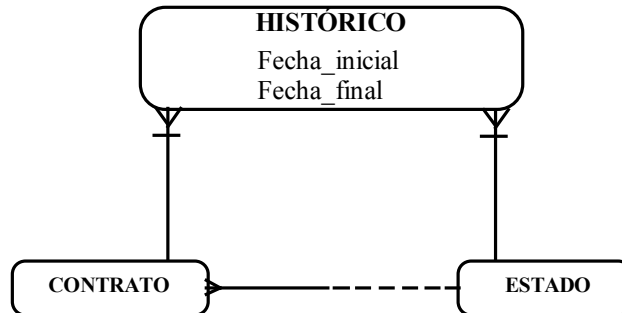
Este es el caso, por ejemplo, de un contrato que puede pasar por varios estados a lo largo de su vigencia.

Siendo de interés guardar información relativa a las fechas en que cada contrato estuvo en un estado particular y a cargo de qué dependencia, se podría modelar un arreglo de atributos consistentes de la tripleta estado/fecha/dependencia, para cada uno de los posibles estados del contrato, como en la figura.

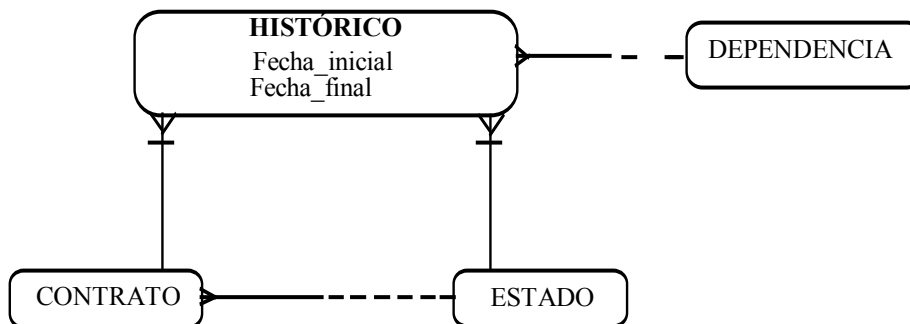


Esta construcción, sin embargo, adolece de varios defectos en la medida en que la mayoría de lenguajes de manipulación de bases de datos (como SQL) no manejan arreglos y, por otra parte, no siempre se siguen los mismos estados de un estado anterior, siendo posible, incluso, que un contrato dado jamás pase por un estado particular.

Una mejor manera de modelar esta situación es crear una entidad llamada ESTADO y definir dos relaciones entre esta y CONTRATO: una (1:N) para el estado actual del contrato y otra (M:N) para la historia de estados pasados, como se ilustra a continuación.

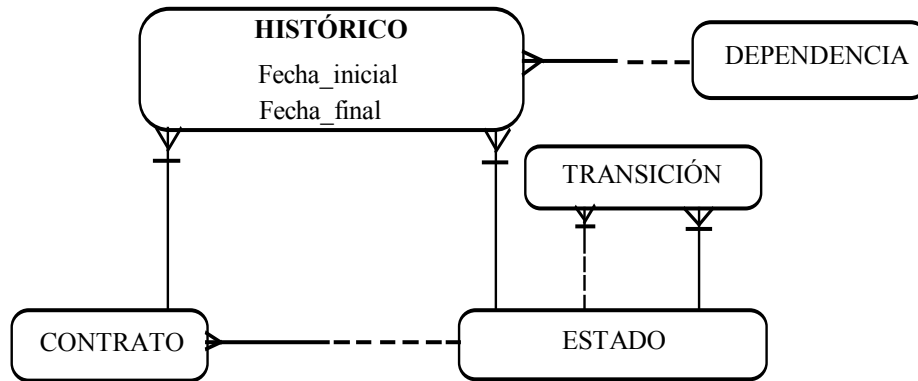


Nótese que la entidad de intersección HISTORICO (que implementa la relación M:N entre CONTRATO y ESTADO) puede, a su vez, relacionarse con la entidad DEPENDENCIA.



Así mismo no es posible pasar de un estado cualquiera a otro, pues el universo de siguientes estados posibles depende siempre del estado actual. Esto define una relación M:N entre ESTADO y ESTADO, representable en la entidad de intersección TRANSICIÓN.

Estos refinamientos se aprecian en el siguiente diagrama.



Una aplicación frecuente de la transición de estados es en el diseño de interfaces de usuario.

6.3 EXPLOSIÓN DE MATERIALES.

Una relación común en las empresas industriales es la de explosión de materiales, en la que se identifica:

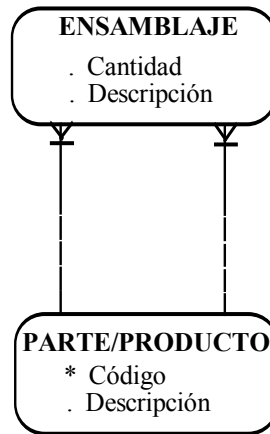
- Qué productos se emplean en la confección de otros productos.
- De qué productos se compone un producto dado.

En este caso, se habla de tipos de partes y productos. El mismo tipo de parte puede emplearse como componente en muchos otros productos y partes. Esto se ilustra a continuación.



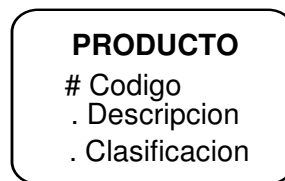
Este modelo resulta insuficiente, pues es necesario conocer, además de qué productos participan en la fabricación de otros, las instrucciones de combinación y la cantidad requerida de cada uno en cada formulación.

Esto se resuelve mediante una entidad de intersección (ENSAMBLAJE), como se ilustra.



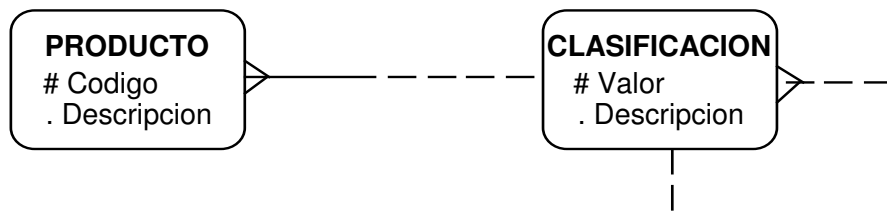
6.4 CLASIFICACIÓN.

Otra construcción común es la de clasificación o categorización de entidades, como en el diagrama de la figura.



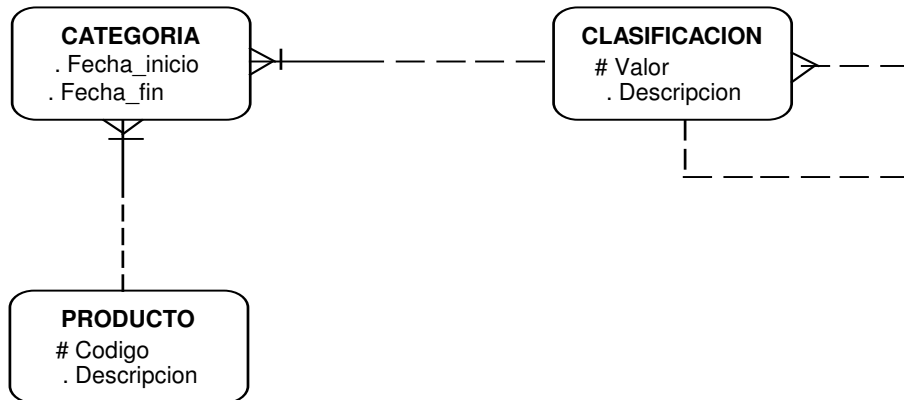
En este caso, se ha empleado un atributo simple para clasificar el producto. Se puede emplear cualquier clasificación para cada producto y se asume que los distintos valores de esta son mutuamente exclusivos, siendo necesario que quien las asigne conozca su significado.

Es posible asociar un significado propio a cada clasificación y (adicionalmente) definir una jerarquía de clasificaciones que detalle completamente la categoría del producto, como en el diagrama.



En este caso, sin embargo, sólo se permite una clasificación del producto. Con frecuencia es necesario clasificar los productos por múltiples conceptos, tales como origen, uso y otras propiedades. Esto se logra reemplazando la relación 1:N existente en el modelo anterior entre clasificación y producto por una relación M:N implementada en una entidad

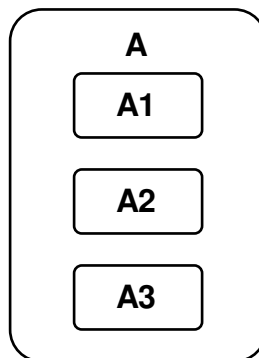
de intersección.



6.5 SUPERTIPOS Y CLASIFICACIÓN.

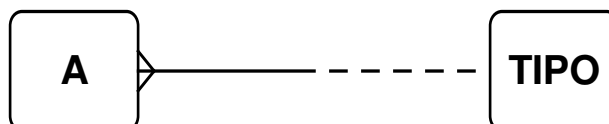
Hay varias maneras de representar supertipos para una entidad, lo que corresponde, en parte, al tema de la clasificación.

Si una entidad A tiene tres subtipos mutuamente exclusivos A1, A2 y A3 se puede representar de la forma convencional como en esta figura.

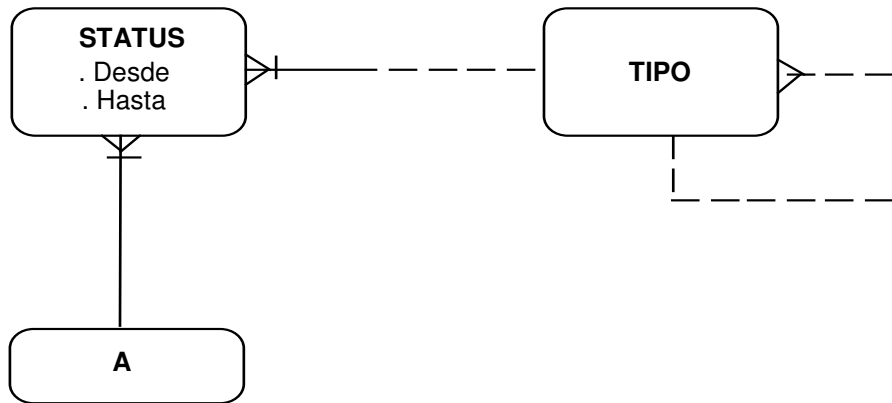


Esta construcción tiene la ventaja de permitir definir atributos y relaciones propias a cada subtipo, pero sólo permite definir exactamente tres subtipos a la entidad A.

Una manera más genérica de manejar tipos de entidad puede ser la creación de una entidad de referencia para cualificar a cada instancia del supertipo.



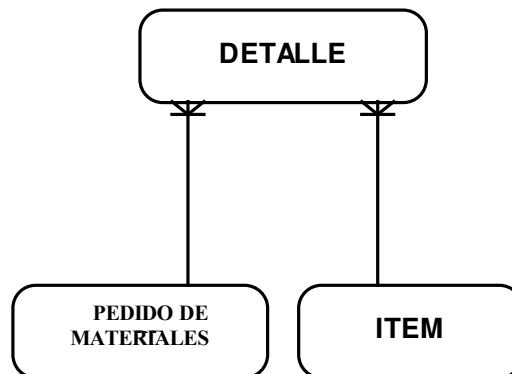
Si un supertipo es ortogonal (esto es, puede constar simultáneamente de varios subtipos) la construcción anterior puede extenderse así:



6.6 TRANSACCIONES ISOMÓRFICAS.

Otra construcción clásica es la de transacciones, que modela la estructura de los documentos manuales que poseen un encabezado y múltiples líneas de detalle.

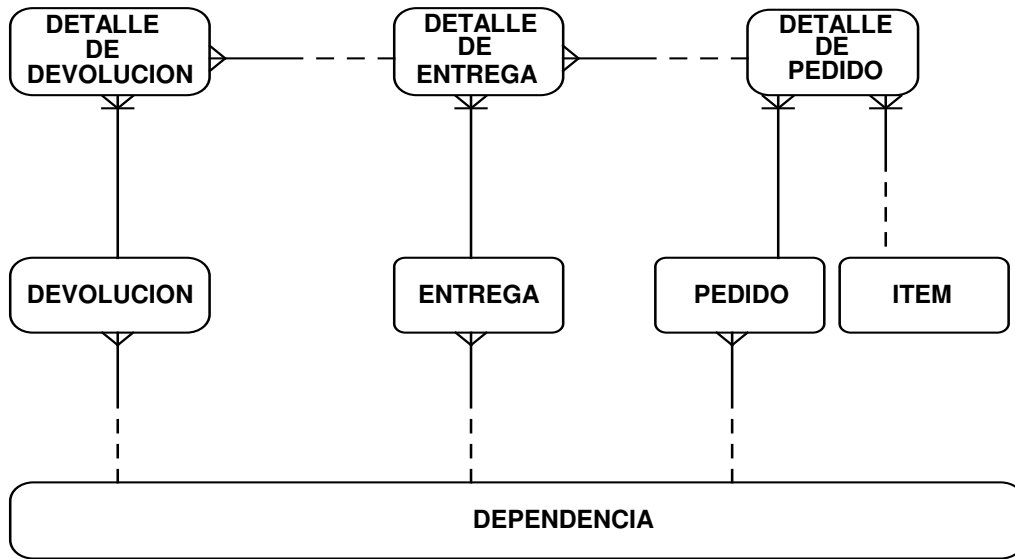
Un ejemplo de esta construcción es un pedido de materiales por parte de una dependencia, como en la figura.



Típicamente, una transacción de este tipo puede dar origen a otra y esta, a su vez, a otra de la misma estructura.

Este es el caso de los pedidos, que pueden originar una o más entregas, las cuales, a su vez, pueden originar una o más devoluciones.

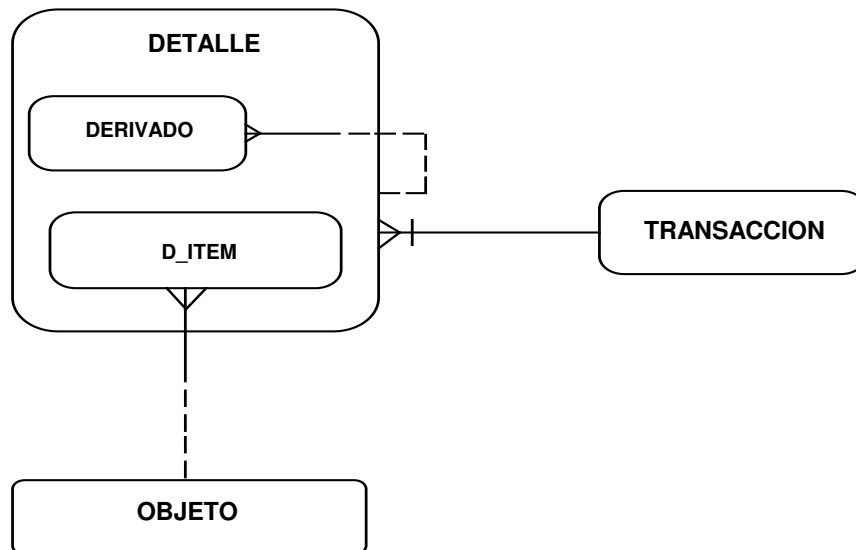
Puesto que sólo es posible entregar lo que ha sido pedido y sólo es posible devolver lo que ha sido entregado, los detalles de entrega referencian a los detalles de pedido y los detalles de devolución referencian a los detalles de entrega, en tanto que únicamente los detalles de pedido referencian directamente al ítem de inventario. Esto se aprecia en el modelo de la figura.



Nótese que esta construcción es válida para las entregas parciales que, por otra parte, son la figura más común en los sistemas de suministro del mundo real.

La redundancia se encuentra, sin embargo, en que todos los encabezados (pedido, entrega y devolución) tienen un número y una fecha y referencian a la dependencia y en que todos los detalles tienen un número de renglón y una cantidad.

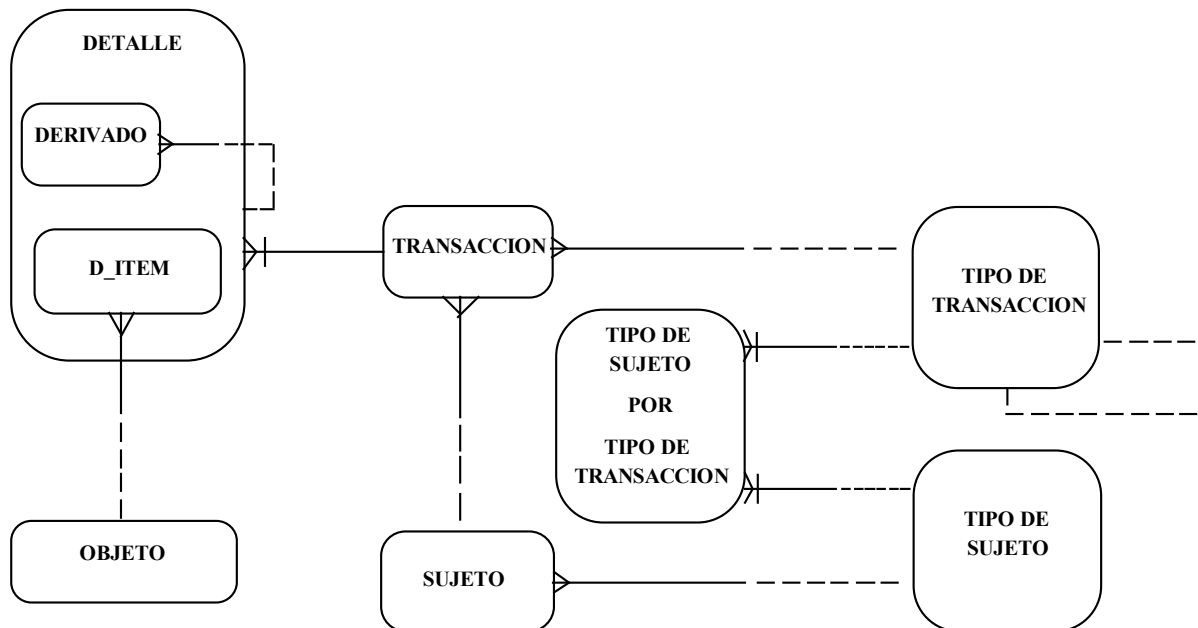
Es evidente que las distintas transacciones (encabezado y detalle) pueden agruparse en dos supertipos, como se ilustra en el diagrama.



Nótese que la entidad DETALLE DE TRANSACCIÓN posee dos subtipos, uno que referencia directamente a ITEM DE INVENTARIO y otro que referencia a un detalle previo de otra transacción.

Esto modela el hecho de que algunas transacciones son originadas por otras anteriores, en tanto que algunas (iniciales) referencian de manera directa a los ítems.

Si se introduce la entidad de referencia TIPO DE TRANSACCIÓN es posible modelar cualquier secuencia de transacciones, como en el diagrama.



Nótese la relación recursiva 1:1 definida sobre TIPO DE TRANSACCIÓN que determina qué transacción puede suceder a otra.

Así mismo, hemos incluido una entidad SUJETO, en reemplazo de la DEPENDENCIA, para permitir distintos tipos de sujetos de transacción tales como proveedor, cliente, etc..

Como construcción complementaria a esta, hemos definido la entidad de referencia TIPO DE SUJETO y una relación M:N entre esta y TIPO DE TRANSACCIÓN para expresar qué tipos de transacción pueden ser originadas por qué tipos de sujeto.

Esto último permite modelar distintos roles jugados por cada categoría de entidad.

Un esquema útil para simplificar el modelaje de este tipo de transacción es considerar que, en general:

- Una PERSONA (natural o jurídica) puede originar una o más TRANSACCIONES, en tanto que una TRANSACCIÓN debe ser originada por una y sólo una PERSONA.
- Una TRANSACCIÓN debe componerse de uno o más DETALLES DE TRANSACCIÓN, en tanto que un DETALLE DE TRANSACCIÓN debe pertenecer a uno y sólo una TRANSACCIÓN.
- Un OBJETO puede ser afectado por uno o más DETALLES DE TRANSACCIÓN, en tanto que un DETALLE DE TRANSACCIÓN debe afectar a uno y sólo un OBJETO.

Para distintos dominios de aplicación particular, la generalización PERSONA puede reemplazarse por CLIENTE, PROVEEDOR o DEPENDENCIA, la generalización TRANSACCION por FACTURA, COMPROBANTE o DESPACHO y la generalización OBJETO por ITEM DE INVENTARIO, CUENTA CONTABLE o CUENTA CORRIENTE.

En todos los casos, se presenta que uno o más atributos del objeto son afectados por la transacción.

Por ejemplo, una transacción de inventarios modifica los atributos existencia y valor total del artículo, un asiento contable afecta los saldos debe y haber de la cuenta contable, una transacción bancaria afecta el atributo saldo de la cuenta corriente.

6.7 TRANSACCIONES RECURRENTE.

En algunas aplicaciones se presentan "moldes" predefinidos en los que los detalles de un tipo dado de transacción se asocian siempre con las misma instancias de objeto.

Este tipo de transacción se conoce como transacción recurrente. Un ejemplo de ello se presenta en la contabilidad general donde (para cada TIPO DE COMPROBANTE) se pueden definir transacciones recurrentes que afectan siempre a las misma cuentas, con el mismo signo (debe o haber) e, incluso, con el mismo valor.

Esta construcción se ilustra en el diagrama entidad/relación del sistema de contabilidad general.

En este diagrama se aprecia que cada TIPO DE COMPROBANTE (de ingreso, egreso, caja, etc.) puede tener varios RECURRENTE que siempre afectan las misma CUENTAS de movimiento.

Así, por ejemplo, la transacción ingresos por venta de útiles de oficina siempre afectará a las cuentas de caja, ventas y comisiones de la misma manera (al debe o al haber).

Esta construcción es muy útil, porque permite al contador definir sus propios mapas de imputación, almacenándolos en la base de datos y no en los programas de aplicación.

Su mayor beneficio, sin embargo, reside en que el programa de captura de asientos contables puede generar automáticamente los detalles de asiento, sin que el operador deba (repetitivamente) especificar las cuentas y los signos, reduciendo así la posibilidad de errores de imputación por omisión de cuentas requeridas o inclusión de cuentas inapropiadas para el tipo de asiento particular.

De esta manera, al insertar un registro en la tabla de cabecera de asiento, el programa de aplicación puede generar automáticamente los detalles a partir de la tabla de inserción 'cuenta por recurrente' y pasar a la forma de detalles ejecutando una operación automática de consulta que despliegue al usuario el único universo de cuentas afectables por el tipo de transacción dado.

La entidad RECURRENTE puede tener un atributo que indique si el tipo de transacción es modificable o no, indicando si el operador está obligado siempre a diligenciar valores para los mismos sin modificar la cuenta o signo.

Este esquema puede extenderse a otras aplicaciones que deban alimentar en forma automática la contabilidad.

Así, por ejemplo, cada transacción de inventarios podría examinar la tabla cuenta por recurrente para determinar las cuentas que deben ser afectadas por el tipo de transacción y generar automáticamente un registro de interface contable que podría aplicarse a la contabilidad en línea o de forma diferida, previa revisión del contador.

6.8 GENERALIZACIÓN.

La mayoría de los patrones estudiados en esta sección del documento se basan en una poderosa construcción del modelaje denominada generalización, mediante la cual es posible reducir modelos muy complejos a otros más simples y manejables.

La principal bondad de esta aproximación, como ya lo hemos señalado, es el independizar las propiedades del modelo de datos de los programas de aplicación.

Por supuesto, es posible llevar la generalización a extremos patológicos en los que no sólo se oscurece el modelo, sino que (contrariamente a lo esperado) se introduce en los programas de aplicación una lógica mucho más compleja y sensible a los cambios de políticas y procedimientos.

7. TRANSFORMACIÓN DEL MODELO E/R AL MODELO RELACIONAL (SQL).

Una de las causas principales del éxito del modelaje entidad/relación estriba en que incluye procedimientos formales de transformación a otros modelos de datos, tales como el relacional, jerárquico, de redes, estructuras de datos en lenguajes de programación y sistemas de archivos convencionales.

En este documento nos ocuparemos únicamente de las reglas de transformación al modelo relacional y, más específicamente al lenguaje SQL empleado por la inmensa mayoría de manejadores de bases de datos relacionales empleados actualmente.

Las reglas de transformación del modelo entidad/relación a modelo relacional garantizan que el esquema resultante se encuentra en tercera forma normal, tal como si se hubiera elaborado con la técnica iterativa de normalización de Codd.

7.1 REGLAS DE TRANSFORMACIÓN.

Las reglas básicas de transformación son:

1. Toda entidad se representa en una tabla separada y sus atributos en columnas de dicha tabla. La clave primaria define un índice no duplicado sobre la(s) columna(s) que la compone(n).
2. Toda relación 1:N se representa replicando la(s) columna(s) componente(s) de la clave primaria de la entidad lado 1 como columna(s) extra de la tabla de la entidad lado N.

Esta replica de la clave primaria del lado 1 en el lado N se denomina clave foránea. Toda clave foránea debería definir un índice duplicado sobre la(s) columna(s) que la conforma(n).

3. Toda relación 1:1 se representa replicando la(s) columna(s) componente(s) de la clave primaria de cada una de las entidades participantes en la relación como columna(s) extra en la tabla de la otra.

Esto se traduce en que se replica del clave primarias de la otra entidad en ambos extremos de la relación. Las columnas de cada clave foránea define un índice no duplicado sobre cada tabla.

Puesto que toda relación M:N ha sido resuelta como una entidad de intersección, se sigue que se debe representar en una tabla separada.

Por otra parte, la clave primaria de una entidad de intersección es siempre la combinación de las claves foráneas correspondientes a las entidades intersectadas.

Así mismo, todas las columnas componentes de la clave primaria y de claves foráneas correspondientes a relaciones mandatorias (debe) deben poseer el atributo NOT NULL.

Como es lógico, las claves foráneas originadas por relaciones opcionales (puede) no deben poseer el atributo NOT NULL.

Aunque los índices no forman parte del modelo lógico, su creación es una consecuencia de la necesidad de hacer accesibles las tablas resultantes del modelo por todas las relaciones definidas en el mismo; esto es, para hacer el esquema navegable.

Como norma general, debería definirse un índice no duplicado sobre la(s) columna(s) componente(s) de la clave primaria. (En algunos manejadores al definir un grupo de columna(s) como PRIMARY KEY automáticamente se crea el índice único y restringe el uso de valores nulos en ella.

Esto tiene como ventajas:

- Garantizar un rápido acceso a cada registro dada su clave primaria. Esta es la operación de acceso más frecuente en aplicaciones transaccionales.
- Garantizar la unicidad de la clave primaria en cada registro.

Las relaciones 1:N deberían definir un índice duplicado sobre la(s) columna(s) correspondientes a la clave foránea en la tabla lado N.

Esto tiene como ventaja permitir un rápido acceso de todos los registros relacionados en la tabla lado N dada la clave primaria de un registro de la tabla lado 1.

Las relaciones 1:1 deberían definir un índice no duplicado sobre la(s) columna(s) correspondientes a las claves foráneas de cada una de las tablas.

Al igual que en el caso de la clave primaria, esto hace más eficiente el acceso a registros relacionados y garantiza la unicidad de la asociación entre los mismos.

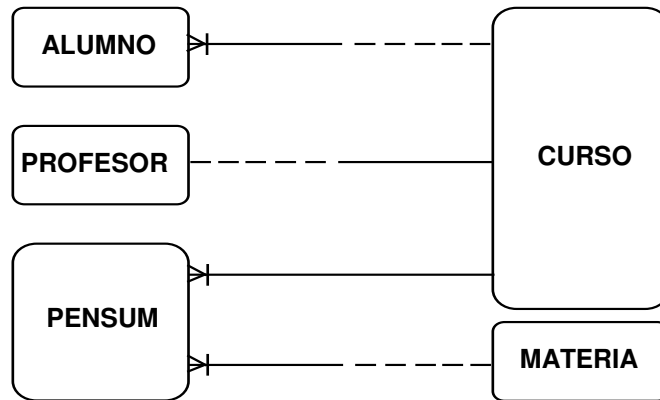
Los índices duplicados derivados de relaciones 1:N pueden o no definirse, dependiendo de si se espera o no efectuar joins frecuentemente por tal concepto.

Si aplicáramos en forma mecánica las reglas anteriores, las entidades de intersección tendrían (cuando menos) tres índices:

- Índice no duplicado (correspondiente a la clave primaria) definido sobre la combinación de las claves foráneas de las entidades intersectadas.
- Índice duplicado correspondiente a la clave foránea de la primera entidad intersectada (redundante!).
- Índice duplicado correspondiente a la clave foránea de la segunda entidad intersectada.

Sin embargo, puesto que el primer componente del índice de la combinación es la réplica de la clave primaria de la primera entidad, es innecesario definir un segundo índice sobre la misma.

Un ejemplo sencillo de transformación de modelo entidad/relación a modelo relacional es el correspondiente al diagrama siguiente:



A partir de él (y considerando los atributos más obvios de cada entidad), tendríamos el siguiente esquema de tablas e índices:

```

CREATE TABLE materia (
    codigo          NUMBER(4)          PRIMARY KEY,
    nombre         VARCHAR2 (16)
);

CREATE TABLE curso (
    codigo          NUMBER(4)          PRIMARY KEY,
    nombre         VARCHAR2(16)      NOT NULL,
    profesor       NUMBER(9)         NOT NULL UNIQUE
    REFERENCES profesor(cedula)
);

CREATE TABLE pensum (
    curso          NUMBER(4)          REFERENCES curso(codigo),
    materia       NUMBER(4)          REFERENCES materia(codigo),
    intensidad    NUMBER(2) ,
    PRIMARY KEY (curso,materia)
);

CREATE INDEX pn_materia ON pensum(materia);

CREATE TABLE profesor (
    cedula         NUMBER(9)          PRIMARY KEY,
    nombre        VARCHAR2(32),
    sexo          VARCHAR2(1)        CHECK ( SEXO IN ('F','M') ),
    salario       NUMBER(9,2),
    curso         NUMBER(4)          UNIQUE
    REFERENCES curso(codigo)
);

CREATE TABLE alumno (
    curso          NUMBER(4)          REFERENCES curso(codigo),
    cedula        NUMBER(9),
  
```

```

nombre          VARCHAR2(32)   NOT NULL,
sexo           VARCHAR2(1)   CHECK (SEXO IN ('F', 'M')),
semestre       NUMBER(2),
PRIMARY KEY (curso,codigo)
);

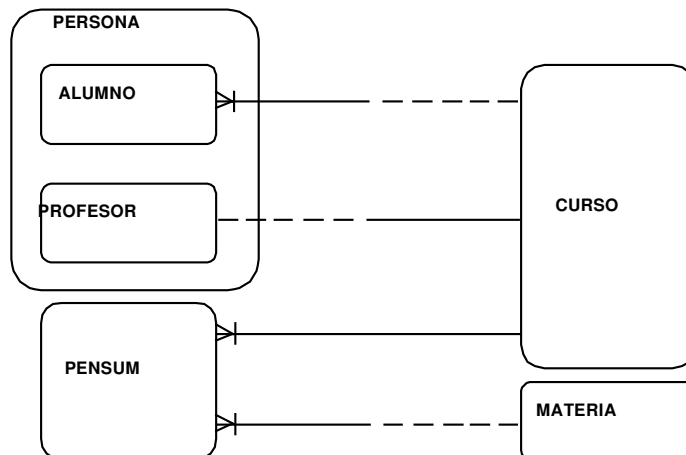
```

Nótese que la clave foránea curso de la tabla alumno no requiere de un índice separado, pues al formar parte de su clave primaria (alumno es una entidad débil respecto a curso) ya ha sido incluido en el índice asociado.

Así mismo, puesto que la relación entre profesor y curso es debe para curso, la clave foránea de profesor en la tabla curso ha sido definida como NOT NULL, a diferencia de la clave foránea de curso en la tabla profesor, cuyo extremo en la relación es puede.

7.2 REPRESENTACIÓN DE SUPERTIPOS.

Las entidades ALUMNO y PROFESOR pueden agruparse en el supertipo PERSONA como se aprecia en el diagrama:



La representación de supertipos en el modelo relacional puede hacerse de dos maneras:

1. Creando una tabla para el supertipo (con los atributos comunes a todos los subtipos) y una tabla separada para cada subtipo, con sus propios atributos.
2. Creando una única tabla que contenga tanto los atributos del supertipo como los atributos de todos los subtipos, empleando vistas parciales para seleccionar las columnas relevantes a cada subtipo.

En el primer caso, las tablas e índices resultantes del anterior diagrama serían:

```

CREATE TABLE curso (
    codigo          NUMBER(4)          PRIMARY KEY,
    nombre         VARCHAR2(16)       NOT NULL,
    profesor       NUMBER(9)           NOT NULL UNIQUE
);

```

```

REFERENCES profesor(codigo)
);

```

```

CREATE TABLE persona (
    cedula          NUMBER(9)          PRIMARY KEY,
    nombre          VARCHAR2(32),
    sexo            VARCHAR2(1)        CHECK (sexo IN ('M','F')),
    tipo            VARCHAR2(1)        NOT NULL
                                CHECK ( tipo IN ('P','A') )
);

```

```

CREATE TABLE profesor (
    persona         NUMBER(9)          PRIMARY KEY
                                REFERENCES persona(cedula),
    curso           NUMBER(4)          UNIQUE
                                REFERENCES curso(codigo),
    salario         NUMBER(9,2)
);

```

```

CREATE TABLE alumno (
    curso           NUMBER(4)          REFERENCES curso(codigo),
    persona         NUMBER(9)          REFERENCES persona(cedula),
    semestre        NUMBER(2),
    PRIMARY KEY(curso,codigo)
);

```

Nótese que el atributo tipo del supertipo PERSONA se define como NOT NULL. Puesto que este atributo es el tipificador del supertipo, sólo podrá tomar los valores P (profesor) y A (alumno).

Así mismo, la clave primaria del supertipo se replica en cada una de las tablas de los subtipos, además será parte de la clave primaria del subtipo propiamente dicha.

Al emplear esta primera aproximación se tiene algunas ventajas:

- Cada subtipo tiene sus propios índices cuando lo requieran y restricciones NOT NULL.
- Los programas de aplicación trabajan únicamente sobre las tablas correspondientes al subtipo de interés.

Pero, también, algunas desventajas:

- El número de tablas del esquema es mucho mayor.
- Si se desea procesar en forma simultánea los atributos del supertipo y cada subtipo, es necesario definir una vista de juntura (join). Puesto que dicha vista referencia más de una tabla, no se pueden efectuar inserciones, actualizaciones o borrados sobre ella.
- Consecuentemente, las operaciones DML INSERT y DELETE deben ejecutarse

doblemente para la tabla del subtipo y la del supertipo.

En la segunda forma de representación de supertipos, resultarían las siguientes tablas, vistas e índices:

```

CREATE TABLE curso (
    codigo          NUMBER(4)          PRIMARY KEY,
    nombre          VARCHAR2(16)      NOT NULL,
    profesor        NUMBER(9)          NOT NULL UNIQUE
    REFERENCES profesor(cedula)
);

CREATE TABLE persona (
    cedula          NUMBER(9)          PRIMARY KEY,
    nombre          VARCHAR2(32),
    sexo            VARCHAR2(1)        CHECK ( sexo IN ('F','M') ),
    tipo            VARCHAR2(1)        NOT NULL
    CHECK ( tipo IN ('P','A') ),
    alumno_curso    NUMBER(1)          REFERENCES curso(codigo),
    salario         NUMBER(9,2),
    semestre        NUMBER(2),
    prof_curso      NUMBER(1)          UNIQUE
    REFERENCES curso(codigo)
);

CREATE VIEW profesor( tipo,cedula,nombre,sexo,curso,salario )
AS SELECT tipo,cedula,nombre,sexo,prof_curso,salario
FROM      persona
WHERE     tipo = 'P';

CREATE VIEW alumno( tipo,cedula,nombre,sexo,curso,semestre )
AS SELECT tipo,cedula,nombre,sexo,alumno_curso,semestre
FROM      persona
WHERE     tipo = 'A';

```

Nótese el uso de prefijos para las columnas de cada subtipo en la tabla base. Esto se hace necesario para evitar ambigüedades tales como el nombre curso empleado tanto en el subtipo PROFESOR como en el subtipo ALUMNO con propósitos y significados diferentes.

Así mismo, nótese que las columnas correspondientes a la clave primaria y al tipificador del supertipo deben replicarse en las vistas de los subtipos, pues su valor se requiere en la inserción de filas para mantener la diferenciación del supertipo y la conectividad de los subtipos.

Esta aproximación tiene algunas ventajas:

- Resultan menos tablas en el esquema.
- Es posible acceder simultáneamente las columnas del supertipo y de los subtipos a través de las vistas.
- Es posible efectuar operaciones INSERT, UPDATE y DELETE directamente sobre la vista porque referencia a una sola tabla.

Y, como es natural, algunas desventajas:

- Todas las columnas de los subtipos deben ser opcionales; esto es, ninguna puede tener el atributo NOT NULL por ejemplo alumno_curso.
- En algunos manejadores de bases de datos, las columnas no empleadas por cada subtipo ocupan espacio de la base de datos.
- En algunos manejadores de bases de datos, los índices no duplicados correspondientes a las claves primarias de los subtipos pueden resultar duplicados para valores NULL.

Otros manejadores de bases de datos relacionales, como Oracle, ofrecen ventajas especiales para esta aproximación de representación de supertipos.

En Oracle, las columnas nulas no ocupan espacio de la base de datos, por lo que no hay desperdicio en cada instancia de subtipo.

Adicionalmente, los índices de Oracle no incluyen aquellas filas en las que las columnas componentes del índice son todas nulas, con lo que el problema de duplicación de claves primarias de los subtipos desaparece.

Por último, Oracle ofrece una facilidad de chequeo de integridad en vistas definidas sobre una única tabla, que se activa mediante la cláusula WITH CHECK OPTION.

De esta manera, es posible especificar que una columna no sea nula, como en el caso de la vista de alumnos ilustrado a continuación:

```
CREATE VIEW alumno( tipo,cedula,nombre,sexo,curso,semestre )
AS SELECT tipo,cedula,nombre,alumno_curso,semestre,nombre,sexo
FROM persona
WHERE tipo = 'A' AND alumno_curso IS NOT NULL
WITH CHECK OPTION;
```

Nótese que, mediante esta cláusula no es posible insertar un registro en la vista alumnos:

1. Si el tipo no es exactamente A.
2. Si el valor de cédula o de curso es nulo.
3. Si el valor de curso no existe en la tabla curso; el reference definido en la tabla base lo garantiza.
4. Si el sexo no es 'M' (masculino) o 'F' (femenino).

7.3 REPRESENTACIÓN DE ARCOS.

Otro aspecto de interés en la conversión de un modelo entidad/relación a modelo relacional es el de representación de arcos o relaciones condicionales.

Existen dos formas básicas de representar arcos en el modelo relacional:

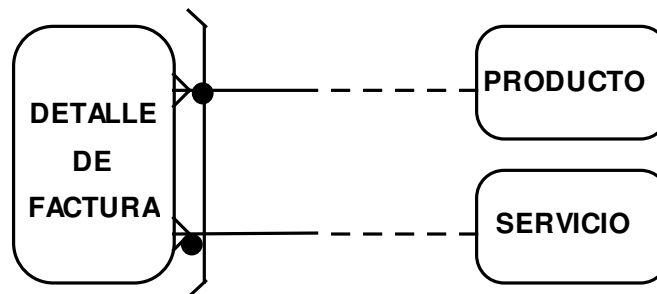
1. Mediante distintas claves foráneas.

2. Mediante conversión de arcos en supertipos.

La primera aproximación consiste en tener tantas claves foráneas como relaciones haya en el arco, cada una con el número de columnas y tipos de datos apropiados para replicar la clave primaria de cada una de las entidades participantes en el arco.

Si todas las claves primarias de las entidades lado 1 en el arco tienen el mismo número de columnas y los mismos correspondientes tipos de datos, es posible crear en la tabla del lado N una única clave foránea que sirva indistintamente a cualquiera de las entidades del arco.

Si (como es común) esta no es la circunstancia, ninguna de las claves foráneas resultantes del arco pueden tener el atributo NOT NULL, pues, dependiendo de la naturaleza de la entidad lado 1 en ocasiones serán alimentadas y en ocasiones no.



Para el ejemplo de arco entre detalle de factura y producto/servicio mencionado anteriormente, la estructura de tablas resultante, según esta primera aproximación sería:

```
CREATE TABLE producto (
    codigo          VARCHAR2(16)    PRIMARY KEY,
    descripción     VARCHAR2(32),
    precio_unitario NUMBER(8, 2),
    unidad          VARCHAR2(3)    NOT NULL
);
CREATE INDEX pd_unidad ON producto(unidad);
```

```
CREATE TABLE servicio (
    codigo          NUMBER(2)      PRIMARY KEY,
    descripción     VARCHAR2(32),
    valor           NUMBER(8, 2)
);
```

```
CREATE TABLE factura (
    numero          NUMBER(4)      PRIMARY KEY,
    fecha           DATE
);
```

```

CREATE TABLE dfactura (
    numero          NUMBER(4),
    renglon         NUMBER(2),
    producto        VARCHAR2(16)
                  REFERENCES producto(codigo),
    cantidad        NUMBER(5, 2),
    servicio        NUMBER(2)
                  REFERENCES servicio(codigo)
                  CHECK ((servicio IS NOT NULL AND producto
                        IS NULL ) OR ( servicio IS
                        NULL AND producto IS
                        NOT NULL )) ,

    PRIMARY KEY(numero, renglon)
);
CREATE INDEX df_producto ON dfactura(producto);
CREATE INDEX df_servicio ON dfactura(servicio);

```

Por supuesto, si las claves primarias de PRODUCTO y SERVICIO pertenecieran al mismo dominio como en:

```

CREATE TABLE producto (
    codigo          NUMBER(3)          PRIMARY KEY,
    ...
CREATE TABLE servicio (
    codigo          NUMBER(3)          PRIMARY KEY,
    ...

```

sería posible definir una única clave foránea en la tabla dfactura, de la siguiente manera:

```

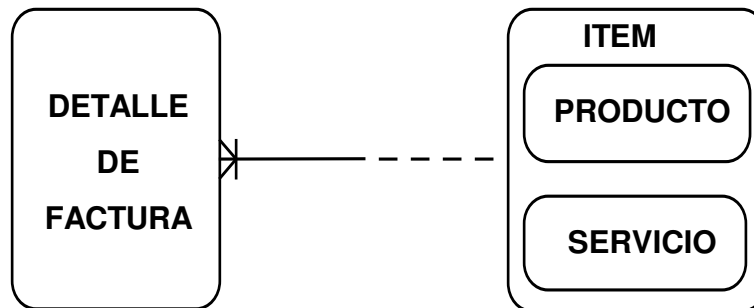
CREATE TABLE dfactura (
    numero          NUMBER(4),
    renglon         NUMBER(2),
    tipo            VARCHAR2(1)        NOT NULL
                  CHECK (tipo IN ('P', 'S')),
    item            NUMBER(3)          NOT NULL,
    cantidad        NUMBER(5, 2),
    PRIMARY KEY(numero, renglon)
);
CREATE INDEX df_item ON dfactura(item);

```

donde tipo indica si el detalle corresponde a un producto (P) o a un servicio (S) e item corresponde a la clave primaria de producto o servicio; lo que se puede garantizar adicionando un trigger a la tabla dfactura, que previo a la inserción de todo registro o a la modificación del atributo item, garantice la existencia de este en la tabla producto ó en la tabla servicio dependiendo del valor de tipo.

La segunda solución (asociar las entidades del arco en un único supertipo) suele ser más apropiada en la mayoría de los casos, siempre que dicho supertipo tenga sentido.

Desde esta perspectiva, las entidades PRODUCTO y SERVICIO se agruparían en el supertipo ITEM, representado en el siguiente diagrama:



Este diagrama se traduce en el siguiente esquema:

```

CREATE TABLE item (
    codigo          NUMBER(3)          PRIMARY KEY,
    descripcion     VARCHAR2(32),
    precio          NUMBER(8, 2),
    unidad          VARCHAR2(3),
    tipo            CHAR(1)            NOT NULL
                                     CHECK ( tipo IN ('P','S')
prod_codigo       VARCHAR2(16)       UNIQUE,
serv_codigo       NUMBER(2)          UNIQUE
);

CREATE INDEX pd_unidad ON item(unidad);

CREATE VIEW producto( tipo,item,codigo,descripcion,precio_unitario,unidad )
AS SELECT  tipo, codigo,prod_codigo,descripcion,precio,unidad
FROM item
WHERE tipo = 'P' AND unidad IS NOT NULL AND prod_codigo IS NOT NULL
WITH CHECK OPTION;

CREATE VIEW servicio( tipo,item,codigo,descripcion,valor )
SELECT tipo,codigo,ser_codigo,descripcion,precio
FROM item
WHERE tipo = 'S' AND serv_codigo IS NOT NULL
WITH CHECK OPTION;

CREATE TABLE factura (
    numero          NUMBER(4)          PRIMARY KEY,
    fecha           DATE
);

CREATE TABLE dfactura (
    numero          NUMBER(4),
    renglon         NUMBER(2),
    item            NUMBER(3)          NOT NULL
                                     REFERENCES item(codigo),
    cantidad        NUMBER(5, 2),
    PRIMARY KEY(numero, renglon)
);

```

CREATE INDEX df_item ON dfactura(item);

8. APÉNDICES.