

3 Modelo relacional

3.1 Introducción

- El modelo de datos relacional fue introducido por Edgar F. Codd en 1970.
- Está basado en una estructura de datos simple y uniforme: la **relación** (en la acepción matemática).
- Tiene un fundamento formal: teoría de conjuntos, lógica de predicados de primer orden, álgebra y cálculo relacionales.
- Formalmente: el acceso a la base de datos se realiza usando lenguajes basados en el álgebra relacional o en el cálculo relacional.
- Los conceptos de este modelo pueden ser usados para crear modelos lógicos de bases de datos, hasta cierto punto independientes de un DBMS particular.
- Es un modelo de datos firmemente establecido en el mundo de las aplicaciones de bases de datos, sobre todo las orientadas a negocios.
- Es la base de una gran cantidad de productos comerciales de DBMS.

3.2 Conceptos

Toda la información de una base de datos (tanto de entidades como de vínculos) se representa por medio de **relaciones** (también conocidas como **tablas**, en el argot comercial). Ejemplo: Mini-sistema escolar:

Prof		
IdProf	NomProf	Categoría
1	Laura	tc
3	Luis	tc
	

Grupo			
ClaveG	Salón	IdProf	ClaveM
300-1	CC102	1	300
310-1	CC001	1	310
		

Alum			
CU	NomAl	Carr	Prom
10	Ana	com	9.3
20	José	eco	9.0
		

Mater		
ClaveM	NomMat	Creds
300	Algoritmos	9
310	Estructuras de Datos	9
	

Inscrito	
CU	ClaveG
10	310-1
10	605-1
....	

Historial				
Folio	Calif	Fecha	CU	ClaveM
1	9	30-may-2016	10	300
2	9	15-dic-2015	20	605
			

Cuando una relación es vista como una **tabla** de valores, cada fila en la tabla representa una colección de datos relacionados. Estos valores pueden ser interpretados como **hechos** que describen una entidad o un vínculo del minimundo.

A continuación se presentan los conceptos principales usados en este modelo:

Dominio

Es un conjunto de valores atómicos. Es el mismo concepto que en el caso del modelo de entidad-vínculo. Ejemplos:

Teléfonos: conjunto de números telefónicos válidos de 10 dígitos;

Nombres: conjunto de cadenas de caracteres de nombres de personas.

Para especificar un dominio normalmente hay que dar un nombre, un tipo de datos y un formato.

Esquema de relación (o esquema relacional)

Está formado por un nombre de relación R y una lista de atributos A_1, A_2, \dots, A_n ; se denota como $R(A_1, A_2, \dots, A_n)$.

Ejemplo: $\text{Prof}(\text{IdProf}, \text{NomProf}, \text{Categoría})$

Cada **atributo** A_i tiene un dominio D asignado y es denotado por $\text{dom}(A_i)$. Un esquema de relación es la *descripción* de una relación.

Relación

Definición informal: una relación debe presentar las siguientes características:

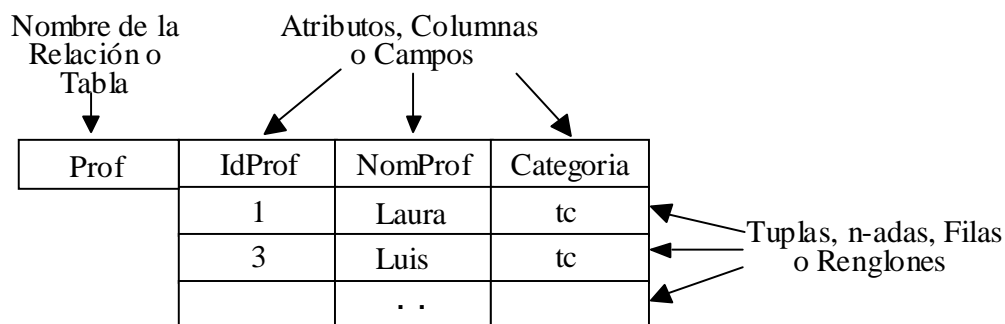
- Las filas contienen datos acerca de una entidad.
- Las columnas contienen datos acerca de atributos de las entidades.
- Todas las entradas en una columna son del mismo dominio.
- Cada columna tiene un nombre único.
- Las celdas contienen valores atómicos.
- El orden de las columnas no es importante.
- El orden de las filas no es importante.
- Dos filas cualesquiera no pueden ser idénticas.

Definición formal: Una relación r del esquema de relación $R(A_1, A_2, \dots, A_n)$ es un conjunto de **tuplas** $r = \{t_1, t_2, \dots, t_m\}$. Cada tupla t es una lista ordenada de n valores $t = \langle v_1, v_2, \dots, v_n \rangle$, donde cada valor v_i es un elemento de $dom(A_i)$ o es un valor **nulo**.

Ejemplo: $r = \{ \langle 1, "Laura", "tc" \rangle, \langle 3, "Luis", "tc" \rangle, \dots \}$

Al número m también se le conoce como *cardinalidad* y al número n como *grado* de la relación.

En la terminología comercial, una relación comúnmente recibe el nombre de **tabla**, un atributo el de **columna** o **campo** y una tupla el de **fila** o **renglón**. Gráficamente:



Esquema de bases de datos relacionales

Un **esquema de base de datos relacional** S es un conjunto de esquemas de relación $S = \{R_1, R_2, \dots, R_m\}$ y un conjunto de **restricciones de integridad** RI . Una **instancia de base de datos relacional** BD de S es un conjunto de relaciones $BD = \{r_1, r_2, \dots, r_m\}$ tal que cada r_i es una relación de R_i y tal que las relaciones r_i satisfacen las restricciones de integridad especificadas en RI .

Ejemplo de esquema de base de datos relacional:

Mini-sistema escolar:

Prof(IdProf, NomProf, Categoría)
Alum(CU, NomAl, Carr, Prom)
Mater(ClaveM, NomMat, Creds)
Grupo(ClaveG, Salón, IdProf(FK), ClaveM(FK))
Inscrito(CU(FK), ClaveG(FK))
Historial(Folio, Calif, Fecha, CU(FK), ClaveM(FK))

Ejemplo de instancia de base de datos relacional:

La figura mostrada al inicio del capítulo para el mini-sistema escolar.

Las restricciones de integridad son especificadas sobre un esquema de base de datos y deben respetarse para cada instancia de ese esquema.

3.3 Restricciones

En esta sección se analizan los diversos tipos de restricciones que pueden especificarse sobre un esquema relacional de base de datos.

Restricción de dominio

Especifica que el valor de cada atributo A de una relación debe ser un **valor atómico** tomado del dominio $dom(A)$ de ese atributo.

Restricción de clave

Dado que una relación está definida como un conjunto, y por definición todos los elementos de un conjunto son distintos, entonces todas las tuplas en una relación deben ser distintas. Esto significa que ningún par de tuplas puede tener la misma combinación de valores para *todos* sus atributos.

Lo anterior da pie a la definición de **clave** de un esquema de relación:

Sea $R(A_1, A_2, \dots, A_n)$ un esquema de relación. Se dice que un subconjunto X de los atributos es **clave** de R si cumple con las siguientes dos propiedades:

- i) *unicidad*: cada valor de X identifica de manera única a cada tupla de la relación;
- ii) *minimalidad*: no se puede remover algún atributo de X y seguir cumpliendo la propiedad (i).

La propiedad de clave debe cumplirse para *cada* instancia de un esquema de relación.

La clave de un esquema de relación es determinada en base al significado de sus atributos.

Un esquema de relación puede tener más de una clave, en cuyo caso cada una se conoce como **clave candidata**. De éstas, una se escoge como **clave primaria**, cuyos valores se usarán para *identificar* a las tuplas de la relación. Normalmente se escogerá como clave primaria aquella que tenga menos atributos o, en caso de empate, aquella que ocupe menos espacio.

Se sigue la convención de subrayar la clave primaria de un esquema de relación para distinguirla o, en ocasiones, ponerle como sufijo: (PK).

Ejemplo:

$$\text{Prof}(\underline{\text{IdProf}}, \text{NomProf}, \text{Categoría})$$
$$0$$
$$\text{Prof}(\text{IdProf(PK)}, \text{NomProf}, \text{Categoría})$$

Restricción de integridad de la entidad

La **restricción de integridad de la entidad** establece que **ningún valor** de la clave primaria puede ser nulo. Esto se debe a que los valores de la clave primaria se usan para identificar tuplas individuales en una relación; si hay valores nulos para la clave primaria no se podrían identificar algunas tuplas.

Restricción de integridad referencial

La **restricción de integridad referencial** (o integridad de referencia) se especifica entre dos relaciones y es usada para mantener la consistencia entre las tuplas de esas dos relaciones.

Para definir formalmente este concepto, primero se definirá el concepto de clave externa:

Un conjunto de atributos FK en un esquema de relación R_1 es una **clave externa** (o clave foránea) de R_1 si satisface las siguientes reglas:

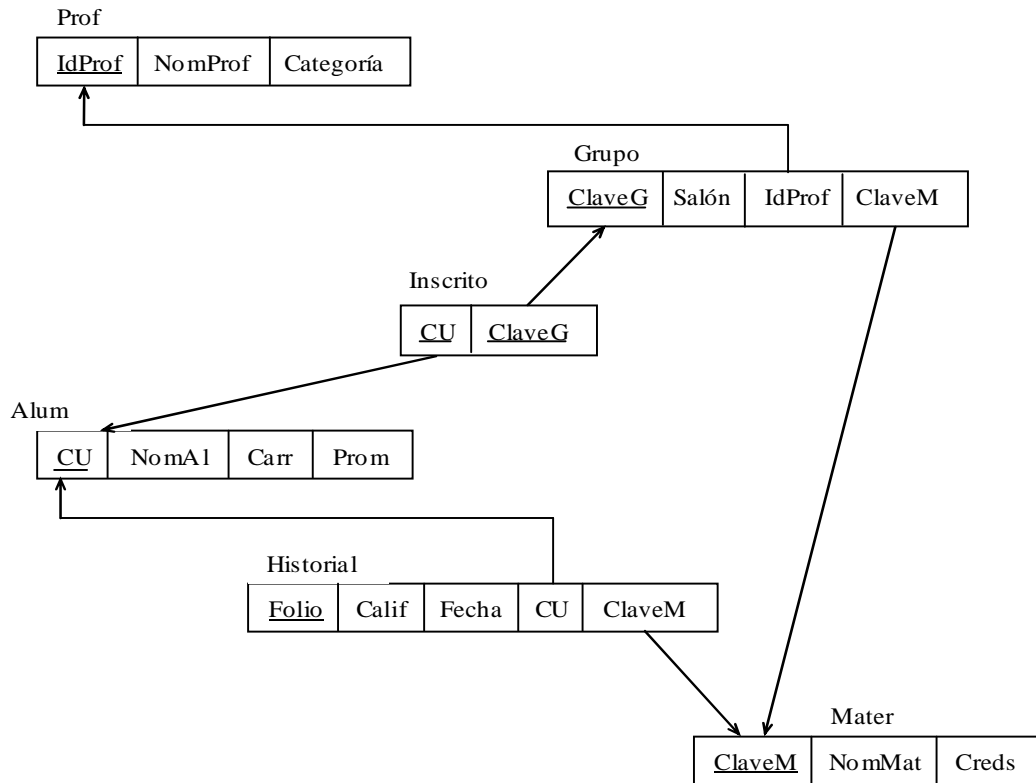
- i) Los atributos en FK tienen el mismo dominio que la clave primaria PK de otro esquema de relación R_2 ; se dice que los atributos FK **hacen referencia** a R_2 (R_2 puede no ser otro esquema, sino el mismo R_1).
- ii) El valor de FK en una tupla t_1 de R_1 debe ser un valor de PK que ya existe para alguna tupla t_2 en R_2 o ser un valor nulo. En el primer caso, se dice que la tupla t_1 **hace referencia** a la tupla t_2 .

Las condiciones anteriores para una clave externa, especifican una **restricción de integridad referencial** entre los dos esquemas de relación R_1 y R_2 .

Ejemplo: El atributo *IdProf* del esquema relacional *Grupo*, es una clave externa que hace referencia a *Prof*. Los atributos *CU* y *ClaveM* del esquema relacional *Historial*, son claves externas que hacen referencia a los esquemas *Alum* y *Mater*, respectivamente.

Un esquema de relación puede tener varias claves externas. Normalmente se distingue a las claves externas de un esquema relacional agregándoles el sufijo: (FK). Para especificar las restricciones de integridad referencial se debe tener un claro entendimiento del significado que cada conjunto de atributos tiene en los esquemas relacionales de una base de datos. Estas restricciones surgen por los *vínculos existentes entre las entidades* representadas por los esquemas de relación.

Si por medio de flechas unimos las claves externas con las claves primarias a las cuales hacen referencia, entonces el siguiente diagrama ejemplifica las restricciones de integridad referencial existentes entre los esquemas relacionales de la base de datos del sistema escolar:



Normalmente un DBMS relacional brinda los elementos necesarios para especificar los diferentes tipos de restricciones comentadas, a fin de mantener la **consistencia (validez) de los datos** almacenados en una base de datos.

3.4 Transformación entidad-vínculo a relacional

En esta sección se describen las reglas que se siguen para transformar un modelo conceptual de una base de datos, del modelo de entidad-vínculo al modelo relacional, a fin de obtener el modelo lógico de ésta, como paso previo a su implementación con un DBMS.

Entidades

Los tipos de entidades simplemente se representan como esquemas de relación (o tablas base).

Ejemplo del mini-sistema escolar:

Prof(IdProf, NomProf, Categoría)
 Alum(CU, NomAl, Carr, Prom)

Cuando un tipo de entidades tiene un **atributo compuesto**, en el esquema relacional sólo aparecerán los **atributos simples (atómicos)** de dicho atributo.

Para **atributos multivaluados** será necesario crear un nuevo esquema relacional por cada atributo, el cual tendrá como clave primaria a los atributos clave del tipo de entidades más el atributo multivaluado.

Vínculos

Se puede crear un nuevo esquema de relación por cada tipo de vínculos existente entre varios tipos de entidades. La clave primaria de estos esquemas estará formada por la conjunción de los atributos clave de los tipos de entidades participantes en el vínculo.

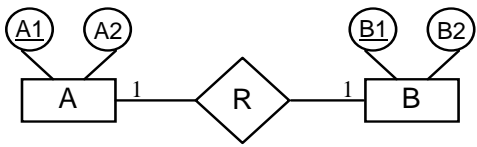
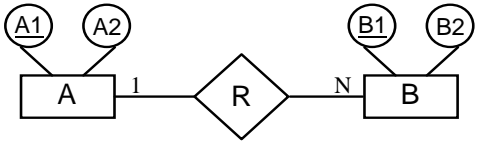
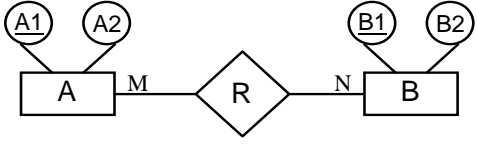
Ejemplo del mini-sistema escolar:

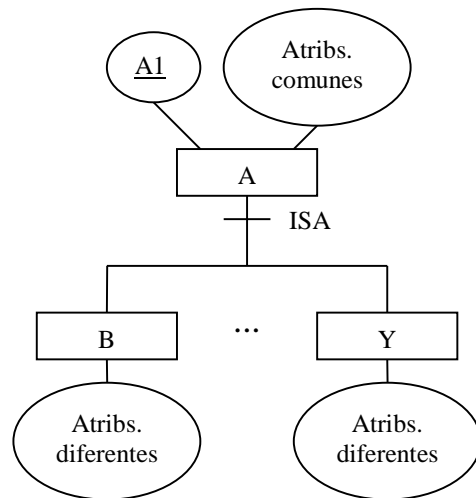
Inscrito(CU(FK), ClaveG(FK))

Hay que observar que en los esquemas relacionales que representan vínculos se aplica la restricción de **integridad referencial**.

Cuando un tipo de vínculos tiene atributos compuestos o multivaluados, se aplican las mismas reglas que en el caso de las entidades.

Cuando la cardinalidad de un tipo de vínculos binarios es 1-1 o 1-N, **normalmente no se crea** un nuevo esquema de relación sino que al esquema de uno de los tipos de entidades participantes se le agrega una clave externa, según se observa en las siguientes figuras:

MODELO DE ENTIDAD-VÍNCULO	MODELO RELACIONAL
	A(<u>A1</u> , A2, B1(FK)) B(<u>B1</u> , B2)
	A(<u>A1</u> , A2) B(<u>B1</u> , B2, A1(FK))
	A(<u>A1</u> , A2) B(<u>B1</u> , B2, A1(FK))
	A(<u>A1</u> , A2) B(<u>B1</u> , B2) R(<u>A1</u> (FK), <u>B1</u> (FK))

Vínculo ISA

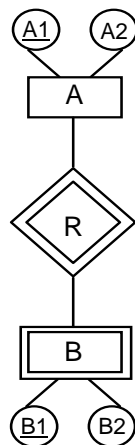
Todos los tipos de entidades, tanto el generalizado como los especializados, generan un esquema relacional por tipo. La clave primaria de todos ellos está formada por la clave primaria del supertipo:

$A(\underline{A1}, \text{Atribs. comunes})$

$B(\underline{A1}(\text{FK}), \text{Atribs. diferentes})$

...

$Y(\underline{A1}(\text{FK}), \text{Atribs. diferentes})$

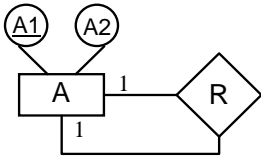
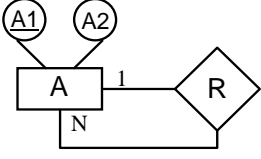
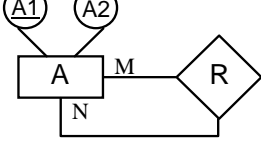
Entidad débil

Cada tipo de entidades se traduce como un esquema de relación. La clave primaria de la entidad débil estará formada por sus propios atributos clave más los atributos de la clave primaria de la entidad propietaria:

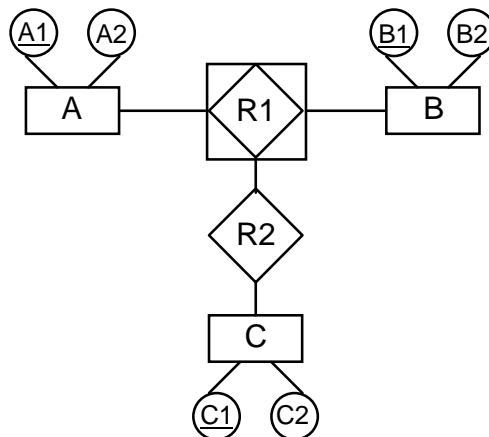
$A(\underline{A1}, A2)$

$B(\underline{A1}(\text{FK}), \underline{B1}, B2)$

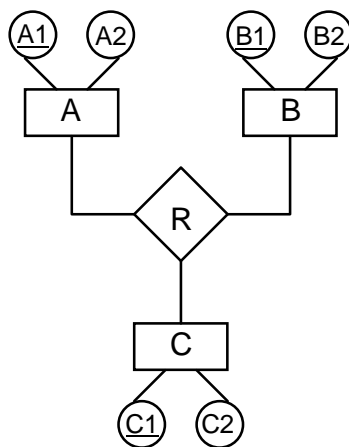
Vínculos recursivos

MODELO DE ENTIDAD-VÍNCULO	MODELO RELACIONAL
	$A(\underline{A1}, A2, A1'(FK))$
	$A(\underline{A1}, A2, A1'(FK))$
	$A(\underline{A1}, A2)$ $R(\underline{A1}(FK), \underline{A1'}(FK))$

En los tres casos, el dominio de A1' es el mismo dominio que el de A1.

Vínculo-entidad

Sin importar la cardinalidad de R1, conviene crear un esquema relacional para el mismo. Después el vínculo con C, a través de R2, ya puede tratarse de acuerdo a la cardinalidad de R2.

Vínculos ternarios

Con cualquier cardinalidad de R, siempre se debe generar un esquema relacional para cada tipo de entidades:

A(A1, A2)

B(B1, B2)

C(C1, C2)

R genera otro esquema; dependiendo de su cardinalidad se escogerá la clave primaria:

a) 1 - 1 - 1

R(A1(FK), B1(FK), C1(FK)) o R(A1(FK), B1(FK), C1(FK)) o
R(A1(FK), B1(FK), C1(FK))

b) 1 - N - 1

R(A1(FK), B1(FK), C1(FK)) o R(A1(FK), B1(FK), C1(FK))

c) 1 - M - N

R(A1(FK), B1(FK), C1(FK))

d) L - M - N

R(A1(FK), B1(FK), C1(FK))

Condicionalidad

Vínculo 1-1c

La clave primaria del lado 1 se coloca como clave externa en el lado 1c. No se permiten nulos para ésta.

Vínculo 1c-N

La clave primaria del lado 1c se coloca como clave externa en el lado N. Se permiten valores nulos para ésta. Si son muchos nulos en la clave externa, es mejor crear un nuevo esquema relacional que represente al vínculo.

Vínculo 1-Nc

La clave primaria del lado 1 se agrega como clave externa en el lado N. No se permiten nulos para ésta.

Vínculo M-Nc

Se crea un nuevo esquema relacional para el vínculo.

Vínculo 1c-1c

Se coloca la clave primaria de un tipo como clave externa en el otro. Se permiten nulos. Si hay mucho nulos, crear una nueva relación con las entidades participantes en el vínculo.

Vínculo 1c-Nc

Ídem 1c-N.

Vínculo Mc-Nc

Ídem M-Nc.

Guía para la traducción de entidad-vínculo a relacional

1. Traducir los tipos de entidad que no tienen clave externa alguna en su esquema relacional correspondiente.
2. Traducir los tipos de entidad que tienen al menos una clave externa.
3. Traducir los vínculos M-N.
4. Traducir los conceptos avanzados.

3.5 Herramientas CASE para modelado

Una herramienta CASE (Computer-Aided Software Engineering) es una herramienta que sirve para auxiliar a los desarrolladores de software en el modelado del mismo. Existen herramientas para elaborar diversos tipos de modelos (flujo de datos, UML, bases de datos, redes, etc.). En particular, algunas de estas herramientas permiten elaborar diagramas de entidad-vínculo usando variantes de la notación de Chen vista en el capítulo 2. En esta sección se describen las características importantes de este tipo de herramientas.

Objetivos

Las herramientas CASE usadas para auxiliar en la elaboración de los diagramas de entidad-vínculo, asociados al desarrollo de bases de datos relacionales, tienen como objetivos principales:

1. Dar los elementos gráficos necesarios para representar los diversos conceptos que se utilizan en la elaboración de los diagramas de entidad-vínculo para modelar la información que será almacenada y manejada en la base de datos.
2. Permitir especificar los tipos de datos de los atributos de las entidades, vínculos, etc., que aparecen en el modelo conceptual de una base de datos concreta. La mayoría de los CASE existentes en el mercado permiten especificar también los diversos tipos de restricciones que pueden observar los DBMS relacionales comerciales.
3. Permitir documentar el objetivo y función del diagrama en general, así como de los tipos de entidades, de vínculos, etc., del modelo conceptual. Esta actividad es importantísima en el desarrollo de las bases de datos la cual, en demasiadas ocasiones, se deja de lado por centrarse más en el modelo conceptual en sí; la herramienta CASE facilita esta tarea de documentación para que el modelado quede más completo.
4. En la mayoría de los casos, permitir generar automáticamente el esquema relacional de la base de datos para un DBMS específico. La mayor parte de los CASE actuales permiten seleccionar el DBMS entre una gran variedad en la cual, normalmente, se incluyen los DBMS's principales del mercado.
5. Dar elementos para efectuar un proceso de *Ingeniería inversa*, el cual consiste en obtener el diagrama de entidad-vínculo a partir de la estructura de una base de datos. Este proceso se puede hacer para varios DBMS, y es de mucha utilidad cuando no se tiene el diagrama para una base de datos dada.

Simbología

Las herramientas CASE normalmente brindan varios tipos de notaciones para la elaboración de los diagramas de entidad-vínculo, incluida la de Chen. La más usada en estos paquetes es una conocida como *notación IDEF* propuesta, en su momento, por el Departamento de Defensa de USA. Tiene varias ventajas sobre las otras notaciones, la más notoria es la de que la representación gráfica de los tipos de entidades y de vínculos que quedan plasmados en un diagrama, tiene gran similitud con la notación que se emplea en el modelo relacional para representar el esquema de una base de datos. También, la generación automática de la estructura de una base de datos, para un DBMS concreto, se hace a partir de un diagrama elaborado con esta notación IDEF.

Aquí cabe aclarar que los conceptos usados en el modelo de datos de entidad-vínculo son **siempre los mismos**, sin importar la notación empleada, ni si el diagrama se hace manualmente o por medio de una herramienta CASE.

La siguiente tabla muestra los conceptos usados en entidad-vínculo y la simbología usada tanto en la notación de Chen, como en la de IDEF:

Concepto	Chen	IDEF
Tipo de entidades	Rectángulo	Rectángulo
Tipo de vínculos	Rombo	Dependiendo de la cardinalidad, puede ser una línea o un rectángulo
Atributo	Elipse	Se escribe dentro del rectángulo
Cardinalidad 1-1		-----● 1
Cardinalidad 1-N		-----● P
Cardinalidad M-N		-----● P Y, además, se crea un nuevo rectángulo para el vínculo
Cardinalidad 1-Nc		-----●
Cardinalidad 1c-N		◇-----● P
Cardinalidad 1c-Nc		◇-----●

Nota: Combinando la simbología anterior se pueden representar en esta notación todos los conceptos de entidad-vínculo.

Microsoft Visio

Es un ejemplo de este tipo de herramientas muy usada en las organizaciones para hacer el modelado de entidad-vínculo.

Comandos iniciales para trabajar con *Visio*:

1. Ventana inicial/Database Model Diagram o File/New/Database/Database Model Diagram.
2. Database/Options/Document/General/Marcar IDEF1X y Conceptual Names. En la pestaña Relationship: Marcar Cardinality y Show Verb Phrase/Forward Text/<Aceptar>.