

# Diseño lógico



- 1.- Introducción
- 2.- Aspectos generales de transformación de esquemas EER a esquemas relacionales
- 3.- Transformación de entidades
- 4.- Transformación de relaciones binarias
- 5.- Transformación de la generalización
- 6.- Transformación de relaciones ternarias
- 7.- Transformación de agregaciones

## 1.- Introducción

En el tema anterior se ha estudiado cómo realizar el diseño conceptual de bases de datos. Para ello, se ha de utilizar un modelo de datos que nos permita obtener un esquema conceptual que represente el mundo real objeto de estudio abstrayéndonos de cualquier aspecto relativo a la posterior implementación (diseño) de la base de datos en un sistema concreto. En nuestro caso, hemos utilizado el modelo EER haciendo uso de la notación definida en la asignatura de Fundamentos de las Bases de Datos.

En la actualidad, no existen Sistema Gestores de Bases de Datos (SGBD) que directamente permitan la definición y manejo de las estructuras tal y como están definidas en el modelo EER. Como ya se ha estudiado en la asignatura de Fundamentos de las Bases de Datos, en la actualidad la mayoría de los SGBD comerciales que nos encontramos están basados en un modelo lógico de datos, heredando el nombre de dicho modelo. Así, podríamos decir que nos podemos encontrar por ejemplo con SGBD en Red (modelo de Red), SGBD jerárquicos (modelo jerárquico) y, SGBD relacionales (modelo relacional). Si bien, en la actualidad prácticamente el 90% de los SGBD comerciales son SGBD relacionales (SGBDR).

Por ello, para poder realizar el diseño final de una base de datos en un SGBD concreto, debemos transformar el esquema EER producido en la etapa de diseño conceptual en un esquema lógico, para su posterior diseño o implementación en un SGBD comercial concreto. Como los SGBD relacionales son los más extendidos, en este tema estudiaremos cómo transformar un esquema conceptual EER en un esquema lógico relacional<sup>1</sup>.

Nos centraremos en la transformación de un esquema EER a un esquema lógico relacional general, dejando para las sesiones de prácticas las transformaciones específicas a un SGBD concreto. El alumno observará, que la mayoría de estas transformaciones siguen una regla “semi-automática”. Sin embargo, existen casos concretos en los que será necesario la interpretación del diseñador para una correcta transformación.

Se invita al alumno a que razone de forma “lógica” estas transformaciones, en lugar de aprenderlas de forma mecánica. Para ayudar a la comprensión de estas reglas, en los primeros casos se expondrán los motivos por los que algunas transformaciones no son válidas. Finalmente, decir que para los ejemplos se utilizarán letras de forma abstracta y, sólo en algunos casos utilizaremos ejemplos concretos.

---

<sup>1</sup> El alumno ya conoce o debe conocer los aspectos generales del modelo lógico relacional estudiados en la asignatura Fundamentos de las Bases de Datos (FBD), por lo que no se estudiarán en detalle en este tema.

## 2.- Aspectos generales de transformación de esquemas EER a esquemas relacionales

En primer lugar introduciremos algunos conceptos que manejaremos en estas transformaciones ya que son soportados por la mayoría de los SGBDR comerciales. De esta forma nos evitaremos realizar puntualizaciones en algunos casos dependiendo de si el SGBDR soporta o no alguna característica específica. Estos conceptos son:

- **Clave candidata:** un atributo (o conjunto de atributos) son clave candidata cuando poseen estas dos propiedades:
  - Unicidad: no existen dos tuplas (filas) con los mismos valores
  - Minimalidad: si quitamos algún atributo de la clave candidata deja de cumplirse la primera propiedad

Una clave candidata puede ser:

- **Clave primaria (C.P.)**
- **Clave alternativa (C. Alt.)**

Tanto si es clave primaria o clave alternativa, partimos de que ninguna de las claves primaria ni alternativas pueden contener nulos, ni valores duplicados.

- **Clave ajena:** un atributo (o conjunto de atributos) que referencia a otra relación R. Este atributo (o conjunto de atributos) bien no tiene valores, o sus valores coinciden con algún valor de la clave primaria de la relación R. (C. Aj.)
- **Valor no nulo (VNN):** las columnas que siempre deben tener valor en cualquier fila de una tabla.

Antes de continuar realizaremos una puntualización respecto a la nomenclatura. El modelo relacional se refiere a relaciones y tuplas para definir sus estructuras (tal y como se ha estudiado en la asignatura FBD). Sin embargo, de manera informal se suelen utilizar los conceptos de tabla y fila para hacer referencia a los conceptos de relación y tupla respectivamente. En este tema utilizaremos ambos términos indistintamente, si bien el alumno ha de tener en cuenta los términos originales definidos en el modelo lógico relacional.

Podríamos resumir las reglas de transformación en tres reglas básicas, que se suelen cumplir con algunas excepciones (por ejemplo, cuando existen atributos multivaluados):

1. Toda entidad se convierte en una relación (tabla).
2. Toda relación N:M se transforma en una relación (tabla).
3. Para toda relación 1:N se realiza lo que se denomina *propagación de clave (regla general, caso más frecuente)*, o se crea una nueva relación. Para la *propagación de clave* utilizaremos el concepto de clave ajena.

Además, en algunos casos particulares, también nos encontraremos con pérdidas semánticas que se comentarán adecuadamente ya que encontraremos algunas situaciones que no tienen representación directa en el modelo relacional. Tal y como se comentó en la introducción, en estas situaciones deberemos acudir a mecanismos propios del SGBD para representarlos y paliar esta pérdida expresiva (definición de procedimientos y funciones en el SGBD, especificación de triggers o disparadores, etc.); si bien, esto último queda fuera de los objetivos básicos de este tema y se abordarán en las clases de prácticas.

En general, para abordar la transformación, la idea es obtener un esquema que, con el menor número posible de tablas, permita recoger todas las restricciones que se contemplan en el esquema conceptual. Aunque, en muchos de los casos que se estudiarán, se pondrá directamente la solución correcta, se debe saber razonar los motivos por los que otras alternativas con menos tablas, no serían correctas para captar el esquema de partida.

### 3.- Transformación de entidades

Una entidad A se transforma en una relación R y los atributos de la entidad A serán atributos de la relación R, donde el atributo identificador de la entidad A será la clave primaria de la relación o tabla R.

A
<u>a0</u>
a1
a2 1..1
a3

A( a0, a1, a2, a3)  
C.P.: a0  
V.N.N.: a2

Se exigen valores no nulos al atributo a2 que se corresponde con el atributo no nulo a2 de la entidad A.

#### 3.1. Atributos multivaluados

Supongamos ahora que tenemos la siguiente entidad A en la que, a diferencia del caso anterior, nos encontramos un atributo multivaluado. En principio, si intentamos seguir utilizando una misma tabla, para que un valor de a0 pueda tener asociados varios valores de a3, pondríamos la clave primaria compuesta (a0,a3).

A
<u>a0</u>
a1
a2 1..1
a3 0..N

A( a0, a1, a2, a3)  
C.P.: (a0, a3)  
V.N.N.: a2

***Incorrecto***

Sin embargo, se puede observar que cada vez que insertemos una tupla para la misma ocurrencia de A (mismo valor de a0), estaríamos repitiendo los valores a1 y a2, con lo que tendremos redundancia de información, por tanto se debería controlar la inserción y modificación de datos, comprobando que se introducen siempre los mismos valores de a1 y a2 para un valor concreto de a0, de otro modo tendríamos información inconsistente.

Podríamos plantearnos evitar la redundancia creando dos relaciones tal y como se realiza a continuación.

A
<u>a0</u>
a1
a2 1..1
a3 0..N

A( a0, a1, a2)	M(a0,a3)
C.P.: a0	C.P.: (a0, a3)
V.N.N.: a2	C.aj.: a0→A

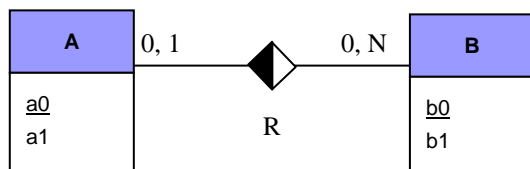
Si se nos indicase que el valor del atributo a3 no se puede repetir para distintos valores del atributo a0, tendríamos que realizar una pequeña variación a esta solución de dos tablas que consiste en definir la clave primaria de la tabla M sólo con la columna a3 (En realidad el EER de partida no era el adecuado, se podía haber mejorado el esquema conceptual)

A( a0, a1, a2)	M(a0,a3)
C.P.: a0	C.P.: a3
V.N.N.: a2	C.aj.: a0→A
	V.N.N.: a0

Podríamos extraer un corolario con respecto a los atributos multivaluados: depende del contexto y de las entidades y atributos particulares la transformación a realizar. Esto se puede aplicar en las transformaciones de atributos multivaluados que veremos más adelante.

## 4.- Transformación de relaciones binarias

### 4.1.- Relaciones 1:N



Podríamos pensar en la transformación más sencilla, que es transformar este esquema en una única tabla. A continuación se muestran tres posibles soluciones y se razona por qué ninguna de ellas puede ser válida:

AB( a0, a1, b0, b1)  
C.P.: a0

**Sol 1**

AB( a0, a1, b0, b1)  
C.P.: b0

**Sol 2**

AB( a0, a1, b0, b1)  
C.P.: (a0,b0)

**Sol 3**

En la sol 1 estamos obligando a que cada vez que introducimos un valor para b0, introduzcamos un valor para a0, con lo que obligamos a que toda ocurrencia de B se relacione con una ocurrencia de A, y el esquema EER del que partimos no tiene representada ninguna restricción de existencia. Además dejamos de tratar a B como entidad, podríamos tener valores en b1, sin tenerlos en b0.

En la Sol. 2 nos encontramos con algunas situaciones por las que no nos sirve esta solución. Por un lado, sucede algo parecido a lo que sucedía en la Sol. 1, pero ahora con los valores de a0; ya que obligamos a que todo valor de a0 esté relacionado con un valor de b0, y esto no está reflejado en el esquema EER. Además, al definir b0 como clave primaria, un valor de b0 no se puede relacionar con más de un valor de a0, con lo que no estamos representando la cardinalidad correctamente. Además dejamos de tratar a A como entidad, pudiendo tener valores de a1 sin tener valores de a0.

Con la sol 3, al poder repetir varias combinaciones de valores para (a0,b0), una instancia/ocurrencia de B se podría relacionar con varias ocurrencias de A y, mientras que según el esquema conceptual, una ocurrencia de A no se debe relacionar con muchas ocurrencias de B. Además, siguen siendo válidos los razonamientos aplicados en los dos casos anteriores en cuanto a las restricciones de existencia que se imponen con la definición de las claves primarias (restricciones de existencia que no están consideradas en el esquema EER). También existiría redundancia, al tener que repetir la misma información para una ocurrencia de B tantas veces como ocurrencias de A con las que se relacionase, y al revés.

Supongamos la siguiente solución alternativa también denominada *propagación de clave*:

A( a0, a1)  
C.P.: a0

B( b0, b1, rA)  
C.P.: b0  
C.aj.: rA → A

Se puede ver que en la relación B no podemos repetir valores para b0, luego una ocurrencia o instancia de B sólo se podrá relacionar con una ocurrencia de A, al revés de lo que se especifica en el esquema EER.

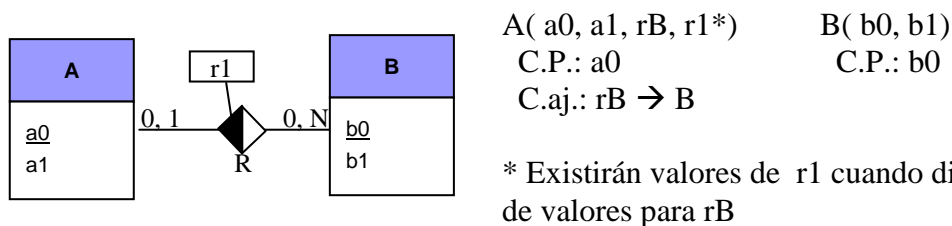
Supongamos ahora la siguiente solución:

A( a0, a1, rB) C.P.: a0 C.aj.: rB → B	B( b0, b1) C.P.: b0
---	------------------------

Ahora sí que se cumple lo que especifica el esquema EER. Como corolario se puede decir que la información sobre la relación en las relaciones 1:N recae sobre la entidad con cardinalidad a muchos (la que está en la parte sombreada de la relación), en lo que se denomina propagación de la clave.

#### 4.1.1.- Atributos de relaciones 1:N

Los atributos de una relación 1:N se definen en la tabla que alberga información sobre la relación. En este caso, se debería añadir una nota a pie de página explicando que no tendría sentido que dicho atributo tuviera valor, si no existe una relación entre dos ocurrencias. Este es uno de los casos en los que es necesario incluir una nota a pie de página pues hay una pérdida semántica al obtener el esquema lógico relacional.



Este es un caso especial, ya que aunque exista esa pérdida expresiva, es la solución que normalmente se suele emplear, a pesar de que para evitarla podríamos hacer

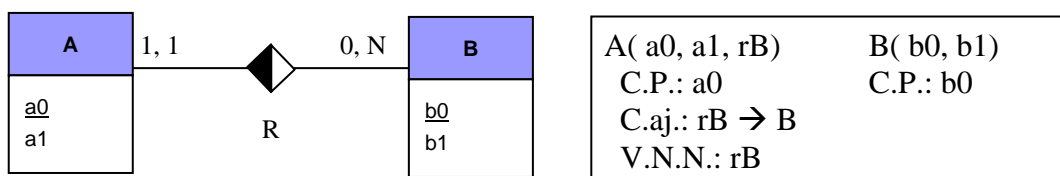
$A(a0, a1)$        $B(b0, b1)$        $R(rA, rB, r1)$   
 C.P.:  $a0$       C.P.:  $b0$       C.P.:  $rA$   
 C.aj.:  $rA \rightarrow A$   
 C.aj.:  $rB \rightarrow B$   
 V.N.N.:  $rB$

Esta solución propuesta es la recoge fielmente el esquema conceptual. Sin embargo, cuando se espera muy frecuentemente valores para  $r1$  al darse la relación  $R$ , se suele trabajar con 2 tablas, tal y como se han definido antes. Esto se hace para ahorrar el control de una clave ajena y el uso de una tabla más en posteriores consultas. Se podría tender a la solución en 3 tablas cuando

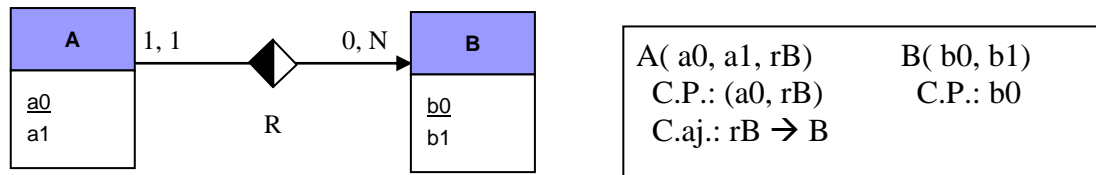
- El número de ocurrencias relacionadas es muy pequeño y, por tanto, existirían muchos valores nulos para la clave ajena según la solución de 2 tablas.
- se tiene una hipótesis de gran peso en que esta relación 1:N se convierta en futuro en una de tipo N:M.
- la relación 1:N tiene un número considerable de atributos propios y no es deseable propagarlos, tanto por conservar la semántica de la relación como por evitar tener muchas ocurrencias con valores nulos para estos atributos.

#### 4.1.2.- Restricciones de Existencia e identificador

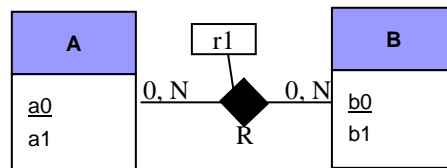
Las restricciones de existencia en las relaciones *1 a muchos* se pueden capturar siempre y cuando la restricción la posee la entidad que posee la cardinalidad a muchos. Dicho de otra forma, sólo se puede capturar una restricción de existencia si la entidad que sufre la restricción, y la relación hacia la que la sufre, se pueden representar en una misma tabla.



En el caso de la restricción de identificador, la clave ajena se incluye en la clave primaria ya que la entidad A necesita de B para su identificación.



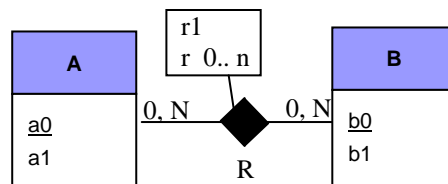
## 4.2.- Relaciones N:M



A( a0, a1)	B( b0, b1)	R(rA, rB, r1)
C.P.: a0	C.P.: b0	C.P.: (rA,rB)
		C.aj.: rA → A
		C.aj.: rB → B

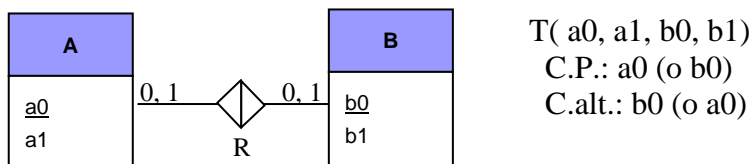
Se deja al alumno que compruebe que no se consigue utilizando sólo dos tablas.

### 4.2.1. Atributos multivaluados en relaciones N:M



A( a0, a1)	B( b0, b1)	R(rA, rB, r1)	R1(rAA,rBB,r)
C.P.: a0	C.P.: b0	C.P.: (rA,rB)	C.P.: (rAA,rBB,r)
		C.aj.: rA → A	C.aj.: (rAA,rBB) → R
		C.aj.: rB → B	

## 4.3.- Relaciones 1:1



**Sol 1: Una sola tabla - Incorrecto**

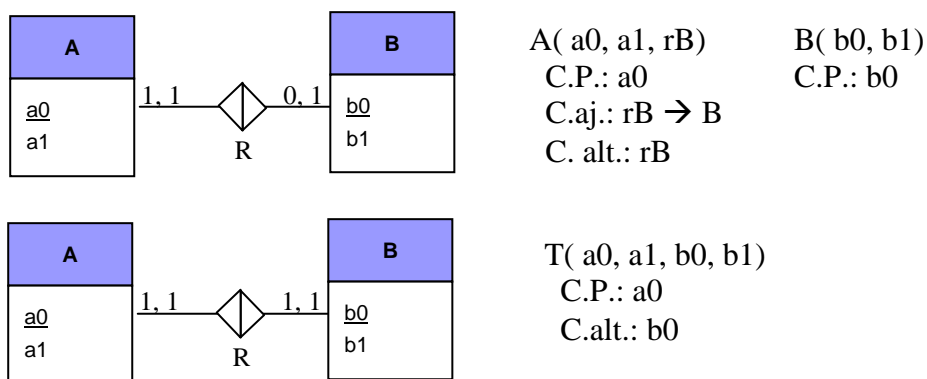
Solución 1: Supongamos que utilizáramos una sola tabla como se muestra anteriormente. Debemos especificar b0 como clave alternativa para respetar la cardinalidad 1:1. Sin embargo, ello implicaría que en cada fila hubiera un valor de b0, con lo que en realidad estaríamos especificando una restricción de existencia, que no aparece en el esquema conceptual.

Solución 2: Se deja al alumno que compruebe que utilizando el concepto de *clave alternativa* no es posible representar este caso con únicamente dos tablas. En el caso de utilizar un SGBD como Oracle que permite la definición de *unique* se podría representar con 2 tablas.

Solución correcta: utilizar tres tablas como se muestra a continuación.

A( a0, a1)	B( b0, b1)	R(rA, rB)
C.P.: a0	C.P.: b0	C.P.: rA
		C.aj.: $rA \rightarrow A$
		C.aj.: $rB \rightarrow B$
		C.alt. rB

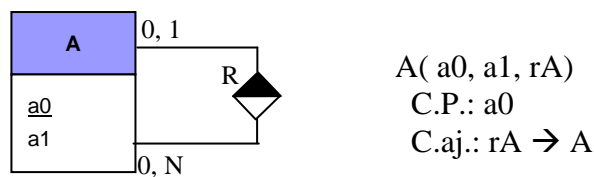
#### 4.3.1.- Restricciones de Existencia



#### 4.4.- Relaciones unarias o reflexivas

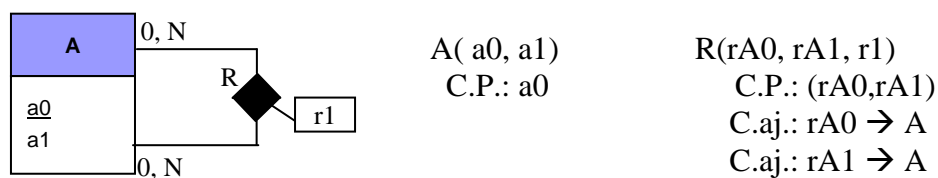
Aunque trataremos estas relaciones, el alumno puede analizarlas como si fueran relaciones binarias, salvo que en este caso sólo interviene una entidad que se relaciona consigo misma.

##### 4.4.1.- Cardinalidad 1:N



La restricción de existencia se capturaría igual que en la relaciones binarias 1:N, especificando valor no nulo en el atributo que conforma la clave ajena. Obviamente, no tiene sentido encontrarnos a nivel conceptual una restricción de identificador en estas relaciones, pues estamos considerando el mismo conjunto de ocurrencias.

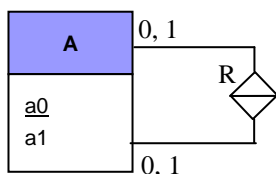
##### 4.4.1.- Cardinalidad N:M





Es fácil comprobar que con una sola tabla no es posible reflejar este esquema.

#### 4.4.1.- Cardinalidad 1:1



A( a0, a1, rA)  
 C.P.: a0  
 C.aj.: rA → A  
 C.alt.: rA

#### Sol 1: Una sola tabla - incorrecta

Es fácil comprobar que la **solución 1** proporcionada anteriormente **no es válida** ya que al especificar rA como clave alternativa estamos obligando a que siempre existan valores para la relación, lo que significaría que existe una restricción de existencia en ambos lados de la relación y esto no es cierto. Luego, esta solución no es válida.

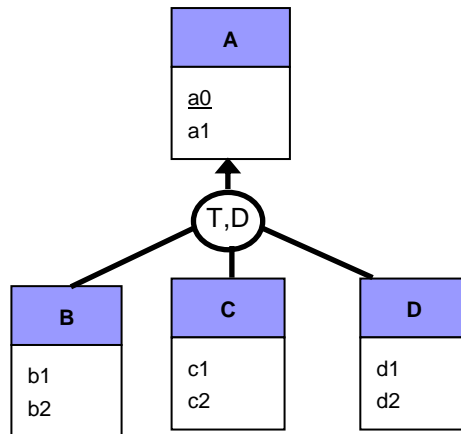
A continuación se proporciona una solución válida para el esquema EER de la figura anterior. (Se deja al alumno comprobar si con Oracle existe otra posibilidad)

A( a0, a1)	R(rA0, rA1)
C.P.: a0	C.P.: rA0
	C.alt.: rA1
	C.aj.: rA0 → A
	C.aj.: rA1 → A

Algunas veces, al utilizar relaciones reflexivas, no queremos que se permita, por ejemplo, la relación de una ocurrencia con ella misma, A1 con A1. Esa es una restricción que no se puede expresar en el esquema EER y, al transformar el esquema al modelo lógico seguimos teniendo el mismo problema. Habría que controlarlo en la entrada de datos.

Otro caso: Si tenemos una relación reflexiva Ser\_amigo; no hay problema en que tengamos las ocurrencias de que Juan es amigo de Luis y, a su vez, Luis es amigo de Juan. Sin embargo, si la relación es Ser\_padre; aquí sí importa el orden pues no es lo mismo ya que si Juan es padre de Luis, obviamente, Luis no puede ser padre de Juan; pero esto no se puede controlar en la transformación ni en el EER. Tendríamos que definir en el SGBD algún disparador (trigger) que controlara esta situación cada vez que se insertara o modificara una fila en la tabla.

## 5.- Transformación de la Generalización



En la generalización se transforma la entidad supertipo en una tabla y, cada una de las entidades subtipo da lugar también a una nueva tabla.

A( a0, a1)	B( b1, b2, bA)	C(c1,cA)	D(d1, d2, dA)
C.P.: a0	C.P.: bA	C.P.: cA	C.P.: dA
	C.aj.: bA → A	C.aj.: cA → A	C.aj.: dA → A

Sin embargo, es fácil comprobar que en realidad en la transformación estamos considerando la generalización Parcial y Solapada. Sería necesario utilizar dos notas a pie de página para comentar que no se pueden representar ninguna de las dos propiedades de la generalización que se muestra arriba. En concreto, no se puede representar que la generalización es:

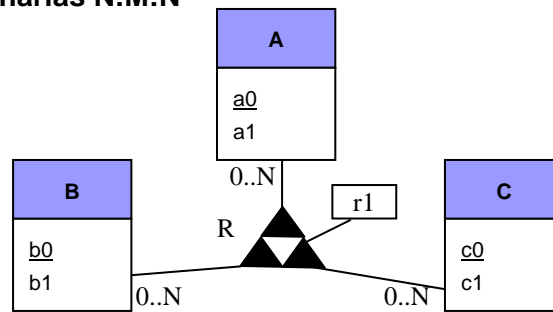
- Total: porque no podemos asegurar que una ocurrencia de A esté necesariamente en A o B o C.
- Disjunta: Las claves primarias de las tablas que representan a los subtipos nos aseguran que no hay duplicados en una misma tabla, pero no contemplan que una ocurrencia que a parece en una tabla no esté en otra de las tablas.

## 6.- Transformación de relaciones Ternarias

El alumno deberá comprobar que, en general, siempre que queramos representar una relación ternaria entre tres entidades diferentes, con menos de cuatro tablas encontraremos redundancias de información. Por lo tanto, utilizaremos al menos cuatro tablas para cualquiera de las cardinalidades de la relación ternaria. Se recuerda que para una correcta transformación debemos prestar especial atención a la cardinalidad máxima expresada por la relación en sí. Ya que utilizaremos una tabla para representar la relación ternaria, independiente de las tablas que representen las entidades, las restricciones de existencias de las entidades del esquema EER hacia la relación ternaria no se pueden representar directamente en un esquema relacional. De nuevo deberíamos recurrir a algún mecanismo en la entrada de datos para controlar esta situación, tal y como se ha comentado anteriormente.

La diferencia básica entre la representación de las ternarias 1:1:1, 1:1:M, 1:M:M o M:M:M reside en la elección de las claves candidatas de la tabla que representa la relación ternaria. Estas claves candidatas se eligen en función de si las tres posibles parejas de ocurrencias, cuya cardinalidad máxima viene marcada en la relación, se pueden repetir o no.

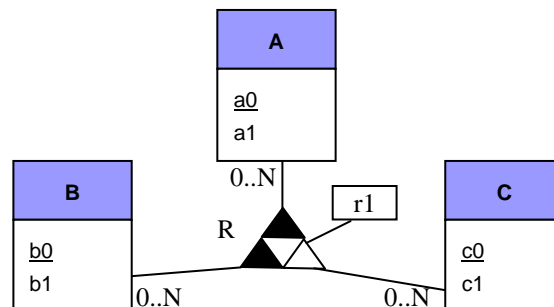
### 6.1.- Relaciones Ternarias N:M:N



A( a0, a1)	B( b0, b1)	C(c0,c1)	R(rA, rB, rC, r1)
C.P.: a0	C.P.: bA	C.P.: c0	C.P.: (rA, rB, rC)
			C.aj.: rA → A
			C.aj.: rB → B
			C.aj.: rC → C

Como cada pareja de ocurrencias se puede relacionar con varias ocurrencias de la otra entidad, es necesario que las tres claves ajenas formen parte de la clave primaria, ya que ninguna de las parejas por sí sola puede ser clave primaria.

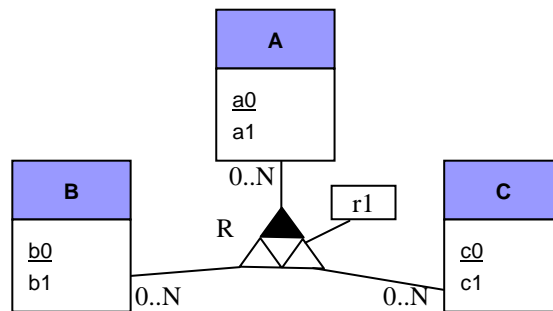
### 6.2.- Relaciones Ternarias N:M:1



A( a0, a1)	B( b0, b1)	C(c0,c1)	R(rA, rB, rC, r1)
C.P.: a0	C.P.: bA	C.P.: c0	C.P.: (rA, rB)
			C.aj.: rA → A
			C.aj.: rB → B
			C.aj.: rC → C
			V.N.N. : rC

Ahora la clave primaria está formada por las claves ajenas de la pareja de entidades (A,B) cada combinación de ocurrencias de A y B como máximo se pueden relacionar con una ocurrencia de la otra (C). La otra clave ajena ha de ser valor no nulo, puesto que si una ocurrencia de A y otra de B se relacionan, siempre tendremos una ocurrencia de C relacionada. Se deja al alumno que razone si se obtiene o no la misma solución si se define rC como clave alternativa.

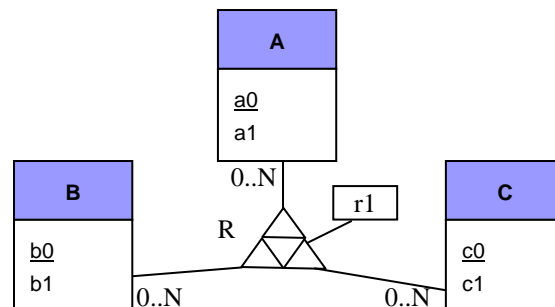
### 6.3.- Relaciones Ternarias M:1:1



A( a0, a1)	B( b0, b1)	C(c0,c1)	R(rA, rB, rC, r1)
C.P.: a0	C.P.: bA	C.P.: c0	C.P.: (rA, rB)
			C.alt.: (rA, rC)
			C.aj.: $rA \rightarrow A$
			C.aj.: $rB \rightarrow B$
			C.aj.: $rC \rightarrow C$

Tenemos dos posibles claves candidatas, las parejas que sólo pueden estar relacionadas con una ocurrencia de la otra entidad. Se elige una como clave primaria y la otra como alternativa.

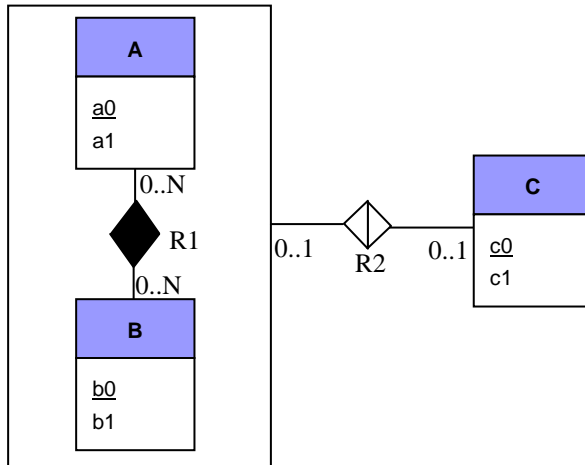
### 6.4.- Relaciones Ternarias 1:1:1



A( a0, a1)	B( b0, b1)	C(c0,c1)	R(rA, rB, rC, r1)
C.P.: a0	C.P.: bA	C.P.: c0	C.P.: (rA, rB)
			C.alt.: (rA, rC)
			C.alt.: (rB, rC)
			C.aj.: $rA \rightarrow A$
			C.aj.: $rB \rightarrow B$
			C.aj.: $rC \rightarrow C$

## 7.- Transformación de relaciones con Agregación

### 7.1.- Relación N:M y relación 1:1

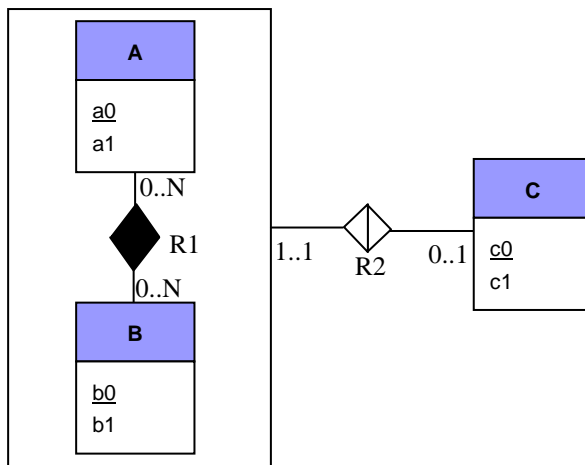


A( a0, a1)    B( b0, b1)    C(c0,c1)  
C.P.: a0      C.P.: b0      C.P.: c0

R1(rA, rB)  
C.P.: (rA, rB)  
C.aj.: rA → A  
C.aj.: rB → B

R2(rC, rA, rB)  
C.P.: rC  
C.alt.: (rA,rB)  
C.aj.: rC → C  
C.aj.: (rA,rB) → R1

#### 7.1.1.- Variante con restricción de existencia



A( a0, a1)    B( b0, b1)    C(c0,c1)  
C.P.: a0      C.P.: b0      C.P.: c0

R1(rA, rB, rC)  
C.P.: (rA, rB)  
C.alt.: rC  
C.aj.: rA → A  
C.aj.: rB → B  
C.aj.: rC → C

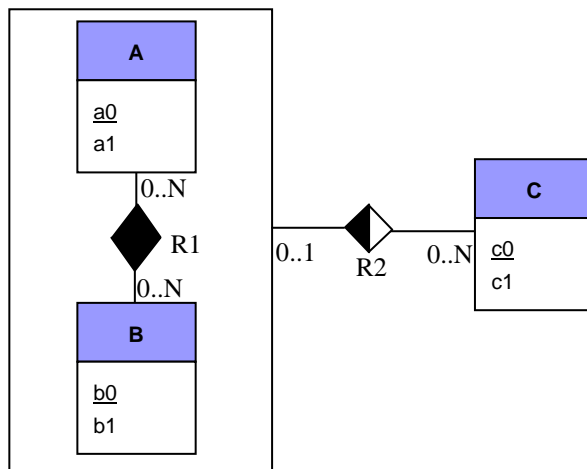
Es fácil comprobar que si la restricción de existencia fuera en la entidad C, en lugar de definir la clave alternativa en R1, la definiríamos en la tabla C y, por tanto, la tabla C quedaría como sigue:

A( a0, a1)    B( b0, b1)  
C.P.: a0      C.P.: b0

C(c0,c1,rA, rB)  
C.P.: c0  
C.alt.: (rA, rB)  
C.aj.: (rA, rB) → R1

R1(rA, rB)  
C.P.: (rA, rB)  
C.aj.: rA → A  
C.aj.: rB → B

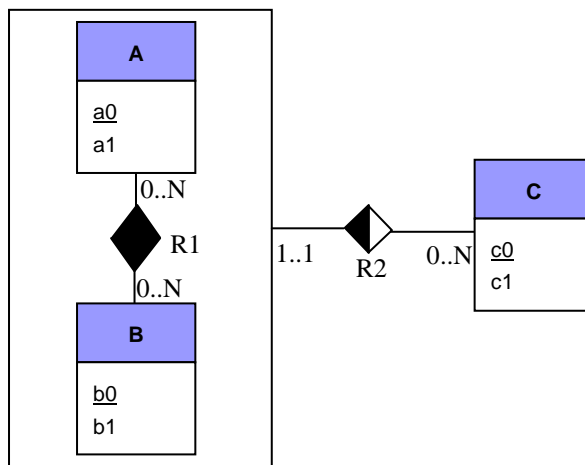
## 7.2.- Relación N:M y relación M:1



A( a0, a1)    B( b0, b1)    C(c0,c1)  
 C.P.: a0      C.P.: b0      C.P.: c0

R1(rA, rB, rC)  
 C.P.: (rA, rB)  
 C.aj.: rA → A  
 C.aj.: rB → B  
 C.aj.: rC → C

### 7.2.1.- Variante con restricción de existencia



A( a0, a1)    B( b0, b1)    C(c0,c1)  
 C.P.: a0      C.P.: b0      C.P.: c0

R1(rA, rB, rC)  
 C.P.: (rA, rB)  
 C.aj.: rA → A  
 C.aj.: rB → B  
 C.aj.: rC → C  
 V.N.N.: rC

Tal y como sucede en las relaciones binarias, si la restricción de existencia estuviera en C no se podría capturar, ya que por motivos de redundancia, la entidad C y la relación R2 no pueden estar en la misma tabla.

Queda propuesto como ejercicio la transformación de otros esquemas con distintas cardinalidades y atributos.