



Diseño lógico de BD y el modelo relacional.

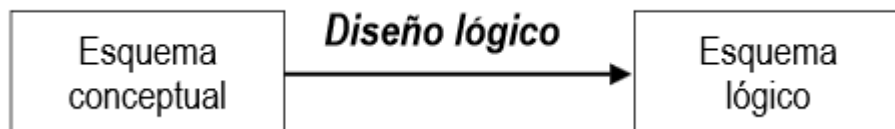
1. Modelo Relacional

El modelo entidad-relación es un modelo conceptual que sirve para cualquier tipo de SGBD, en cambio, el modelo relacional es un modelo lógico que sólo sirve para SGBD relacionales (y no para jerárquicos, como por ejemplo Codasyl).

Todos los diseñadores y administradores de bases de datos relacionales usan esquemas conceptuales entidad-relación porque se adaptan muy bien a este modelo.

Hay que tener en cuenta la diferencia de la palabra relación en ambos modelos. En el modelo relacional una relación es una tabla mientras que en el entidad/relación es la asociación que se produce entre dos entidades.

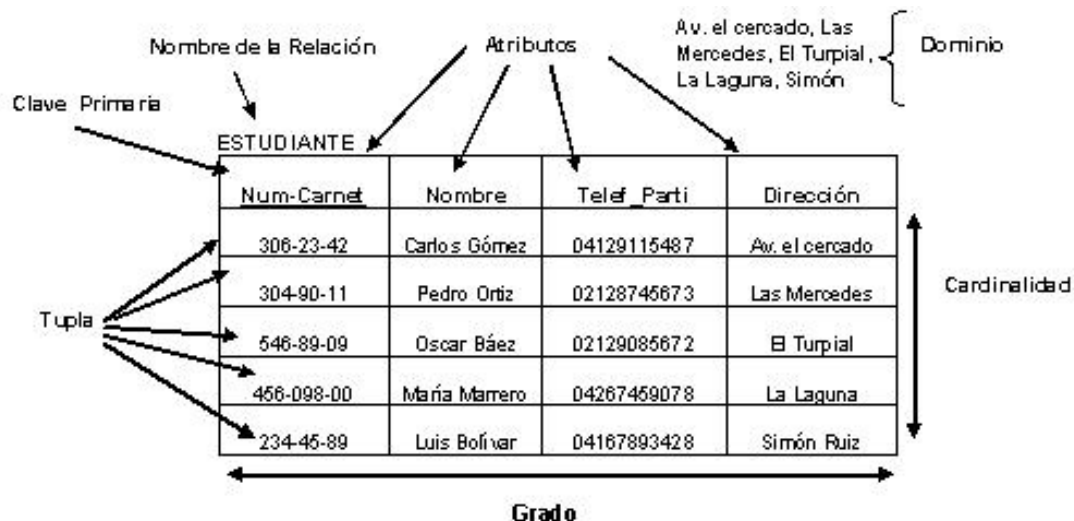
En este tema nos vamos a centrar en convertir el Esquema Conceptual (para nuestro caso el EER) en un esquema lógico (modelo relacional).



1.1. Conceptos

1.1.1. Relación (tabla)

Según el modelo relacional el elemento fundamental es lo que se conoce como relación, aunque habitualmente se le llama tabla. Se trata de una estructura formada por filas y columnas que almacena los datos referentes a una determinada entidad o relación del mundo real.



Acerca de una tabla, además de su nombre, podemos distinguir lo siguiente:

- Atributo

Representa una propiedad que posee esa tabla. Equivale al atributo del modelo E-R. Se corresponde con la idea de campo o columna.

- Tupla

Cada una de las filas de la tabla. Se corresponde con la idea de registro. Representa por tanto cada elemento individual (ejemplar, ocurrencia) de esa tabla.

- Dominio

Un dominio contiene todos los posibles valores que puede tomar un determinado atributo. Dos atributos distintos pueden tener el mismo dominio. Un dominio en realidad es un conjunto finito de valores del mismo tipo. Los dominios poseen un nombre para poder referirnos a él y así poder ser reutilizable en más de un atributo.

- Grado

Número de columnas de la tabla (número de atributos).

- Cardinalidad

Número de tuplas de una tabla (número de filas).

1.1.2. Clave

- Clave candidata

Conjunto de atributos que identifican unívocamente cada tupla de la relación. Es decir columnas cuyos valores no se repiten para esa tabla.

- Clave primaria

Clave candidata que se escoge como identificador de las tuplas. Se elige como primaria la candidata que identifique mejor a cada tupla en el contexto de la base de datos. Por ejemplo un campo con el DNI sería clave candidata de una tabla de clientes, aunque si en esa relación existe un campo de código de cliente, este sería mejor candidato para clave principal, porque es mejor identificador para ese contexto.

- Clave alternativa

Cualquier clave candidata que no sea primaria.

- Clave externa, ajena o foránea

Atributo cuyos valores coinciden con una clave candidata (normalmente primaria) de otra tabla.

- Valores nulos

Las bases de datos relacionales permiten más posibilidades para el valor nulo (null), aunque su significado no cambia: valor vacío. Es importante indicar que el texto vacío ' ', no significa lo mismo en un texto que el nulo; como tampoco el valor cero significa nulo.

1.1.3. Restricción

Igual que en otros modelos de datos, en el Modelo Relacional existen restricciones, es decir, estructuras u ocurrencias no permitidas. Los datos almacenados en la base de datos tienen que adaptarse a las estructuras impuestas por el modelo y deben cumplir una serie de reglas para garantizar que son correctos.

Estas restricciones pueden ser de dos tipos fundamentales: restricciones inherentes, que son impuestas por el propio modelo, y restricciones de usuario (también denominadas restricciones semánticas) en las cuales es el usuario quien prohíbe, porque el modelo se lo permite, determinadas circunstancias.

1.1.3.1. Restricciones inherentes al modelo.

- No puede haber dos tuplas iguales
- El orden de las tuplas no es significativo
- El orden de los atributos no es significativo
- Cada atributo sólo puede tomar un valor en el dominio en el que está inscrito

1.1.3.2. Restricciones semánticas o de usuario.

- Clave primaria (primary key)

Hace que los atributos marcados como clave primaria no puedan repetir valores. Además obliga a que esos atributos no puedan estar vacíos. Si la clave primaria la forman varios atributos, ninguno de ellos podrá estar vacío.

- •Unicidad (unique)

Impide que los valores de los atributos marcados de esa forma, puedan repetirse. Esta restricción debe indicarse en todas las claves alternativas.

- •Obligatoriedad (not null)

Prohíbe que el atributo marcado de esta forma no tenga ningún valor (es decir impide que pueda contener el valor nulo, null).

- •Integridad referencial (foreign key)

Sirve para indicar una clave externa. Cuando una clave se marca con integridad referencial, no se podrán introducir valores que no estén incluidos en los campos relacionados con esa clave. Esto último causa problemas en las operaciones de borrado y modificación de registros, ya que si se ejecutan esas operaciones sobre la tabla principal quedarán filas en la tabla secundaria con la clave externa sin integridad.

Hay sistemas gestores de bases de datos que permiten acciones distintas para el caso de borrado y de actualización de la clave.

Las posibilidades son las siguientes:

- Borrado y/o modificación en cascada (CASCADE). El borrado o modificación de una tupla en la relación padre (relación con la clave primaria) ocasiona un borrado o modificación de las tuplas relacionadas en la relación hija (relación que contiene la clave ajena).

- Borrado y/o modificación restringido (RESTRICT). En este caso no es posible realizar el borrado o la modificación de las tuplas de la relación padre si existen tuplas relacionadas en la relación hija.
- Borrado y/o modificación con puesta a nulos (SET NULL). Esta restricción permite poner la clave ajena en la tabla referenciada a NULL si se produce el borrado o modificación en la tabla primaria o padre.
- Borrado y/o modificación con puesta a valor por defecto (SET DEFAULT). En este caso, el valor que se pone en las claves ajenas de la tabla referenciada es un valor por defecto que se habrá especificado en la creación de la tabla.

Los sistemas gestores más avanzados, más potentes, permiten otras restricciones consistentes en comprobar una determinada condición después (o antes) de una actualización.

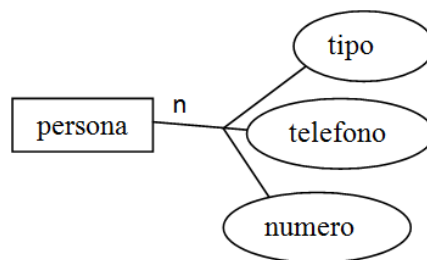
1.2. Paso de Entidad-Relación al modelo Relacional

Previo a la aplicación de las reglas de transformación de esquemas entidad-relación a esquemas relacionales es conveniente la preparación de los esquemas entidad-relación mediante la aplicación de unas reglas que faciliten y garanticen la fiabilidad del proceso de transformación. Estas reglas preparatorias se basan en la aplicación de la 1FN y su objetivo es eliminar las siguientes anomalías:

ELIMINACIÓN DE ATRIBUTOS MÚLTIPLES

Todos los atributos múltiples se deben transformar en un tipo de entidad débil por existencia con una relación de muchos a muchos o de uno a muchos, según sea el caso, con el tipo de entidad sobre el cual estaba definido. Si se considera que la nueva entidad creada resulta ambigua, se le pueden añadir atributos o heredarlos de la otra entidad.

Suponemos para el siguiente ejemplo que una persona puede tener varios números de teléfono.



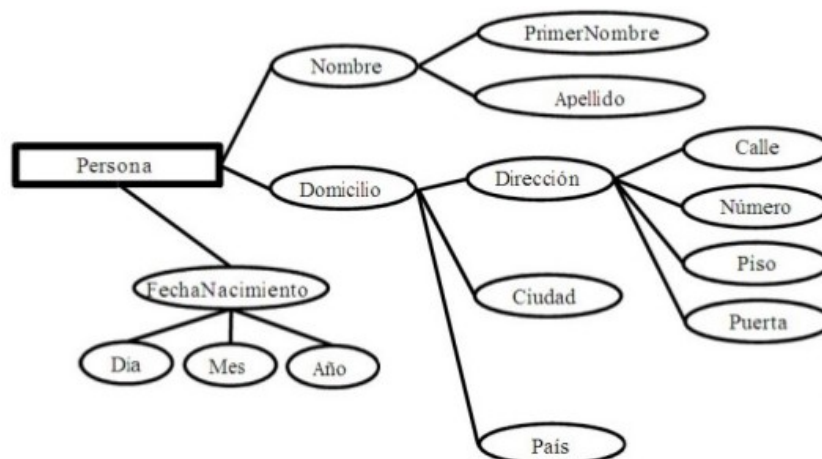
En este caso sería necesario expresar el esquema relacional de la forma:

PERSONA (... lista de atributos ...)

TELEFONO(idPersona*, numero, tipo)

ELIMINACIÓN DE ATRIBUTOS COMPUESTOS

Todos los atributos compuestos deben ser descompuestos en atributos simples que quedan asociados a la misma entidad. El esquema entidad-relación:



Se expresaría mediante el siguiente esquema relacional:

PERSONA (PrimerNombre, Apellido, calle, número, piso, puerta, ciudad, país, día, mes, año)

1.3. Transformación de objetos conceptuales en objetos relacionales

Su objetivo es convertir el modelo conceptual, representado mediante un diagrama E/R en un modelo Lógico para el modelo relacional (SGBD o ficheros indexados) que es una representación en la que las entidades, atributos y relaciones del M-E/R se transforman en tablas, columnas, claves primarias y claves ajenas, aplicando las Reglas de transformaciones siguientes:

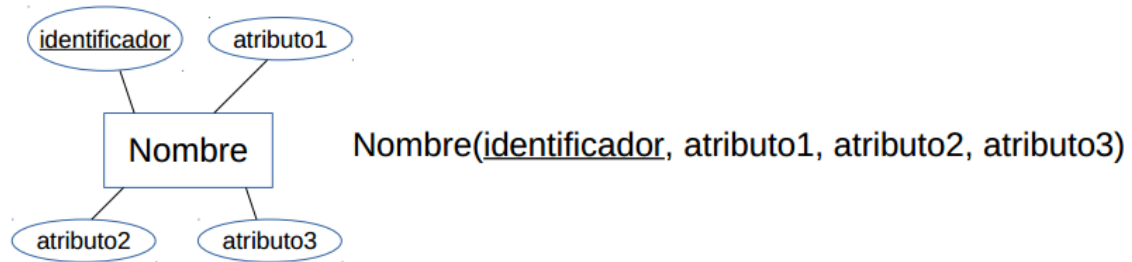
1.3.1. Transformación de las entidades fuertes

En principio las entidades fuertes del modelo E-R son transformadas al modelo relacional siguiendo estas instrucciones:

- **Entidades:** Las entidades pasan a ser tablas.
- **Atributos:** Los atributos pasan a ser columnas.
- **Atributo identificador:** Pasan a ser clave principal de la tabla.
- **Identificadores candidatos:** Pasan a ser claves candidatas.

Esto hace que la transformación se produzca según este ejemplo:

PK: identificador



PK: Primary Key
(Clave primaria)

FK: Foreign Key
(Clave Ajena)

UK: Unique Key
(Clave alternativa)

VNN: Value Not Null
Valor no nulo

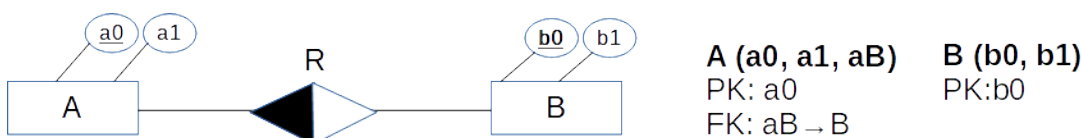
En la figura anterior se puede ver que la entidad, llamada **Nombre** pasa a ser una tabla. A su vez, el identificador (clave primaria de la entidad), pasa a ser la clave primaria de la tabla. Los atributos de la entidad Nombre pasan como columnas normales a la tabla **Nombre**.

Este procedimiento podríamos decir que es el procedimiento estándar para pasar del modelo conceptual al lógico una cierta entidad. A continuación veremos las formas de como pasar las entidades que se relacionan entre sí, que al haber relaciones por en medio el paso al modelo lógico no es tan trivial como el caso anterior.

1.3.2. Transformación de relaciones

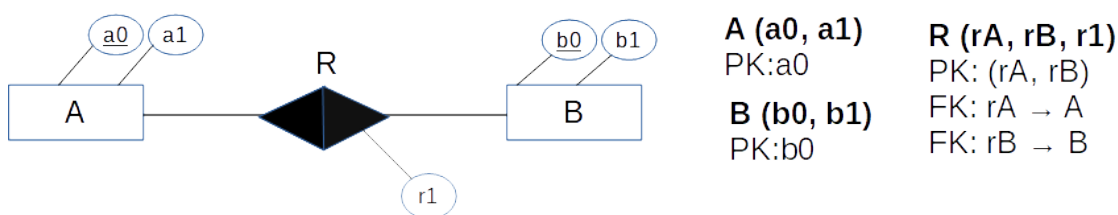
La idea inicial es transformar cada relación en una tabla, pero hay que distinguir según el tipo de relación.

- 1) Toda relación 1:N provoca el fenómeno de propagación de clave, que significa que, el atributo que forma la clave principal de la tabla cuya cardinalidad máxima es 1, debe ser clave ajena en la tabla cuya cardinalidad máxima es N, y por tanto debe añadirse en esta última como columna



En la figura podemos ver como la entidad B queda como una tabla B con las columnas b0 y b1, donde b0 es su Clave primaria (PK). La entidad A queda como una tabla A con las columnas a0 y a1, donde a0 es su clave primaria(PK), además tiene una clave ajena (FK) de aB a la tabla B.

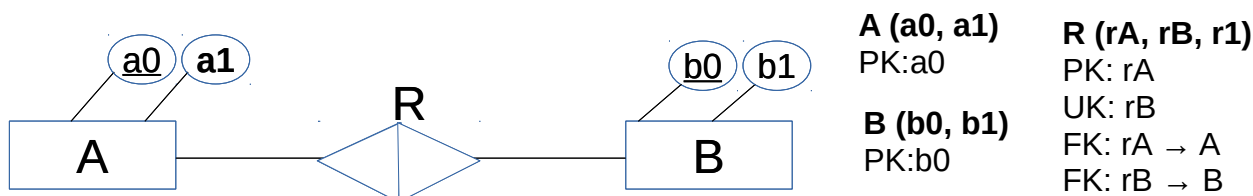
- 2) Toda relación N:N se transforma en una tabla formada por los atributos de la relación, y cuya clave principal será la concatenación de las claves principales de las tablas con las que se relaciona que actúan, además, como claves ajenas. Es decir, se produce una doble propagación de clave



En la figura podemos ver que la entidad A queda transformada al modelo relacional como una Tabla A con los atributos a0 y a1 en forma de columna de esa tabla, además a0 será la clave primaria (PK:a0). Lo mismo ocurre para la entidad B, quedando transformada del mismo modo. Además, la relación R queda transformada como una tabla llamada R formada por la clave primaria de A y de B, en este caso es rA y rB. Estos dos atributos pasan a ser la clave primaria de R. Además, la relación R tiene una clave ajena (FK) desde rA a la tabla A, y otra clave ajena desde rB a B.

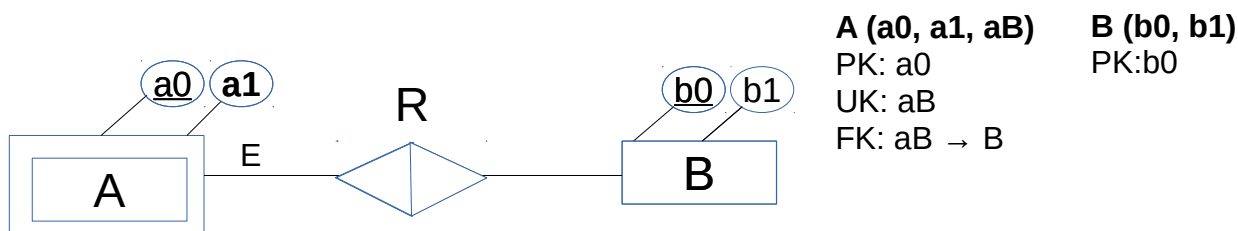
1.3.2.1. Relación 1:1

Si las entidades poseen cardinalidades (0,1), el modelo relacional se queda como tres tablas, una tabla A, otra tabla B y la relación R como tabla R que contendrá como columnas la clave principal de la entidad A (a0), que la hemos llamado rA, la clave principal de la entidad B (b0), que la hemos llamado rB y los atributos de la relación si los tuviera. Además la clave primaria de R será rA, siendo rB clave alternativa de R. A su vez contendrá una clave ajena desde rA a la tabla A y desde rB a la tabla B.

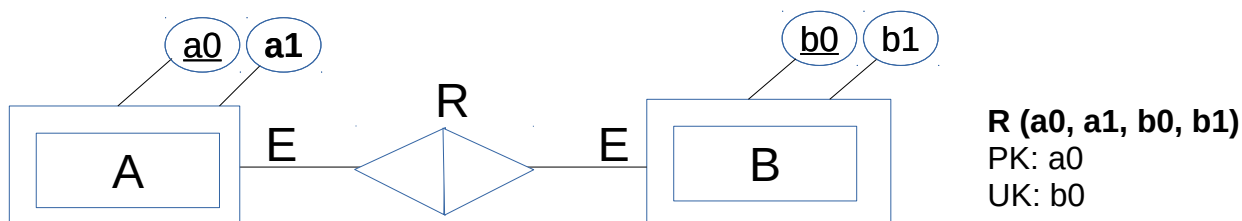


Si hay restricción de Existencia

Si una de las entidades posee cardinalidad (0,1) y la otra (1,1), conviene propagar la clave de la entidad con cardinalidad (1,1) a la tabla resultante de la entidad con cardinalidad (0,1).



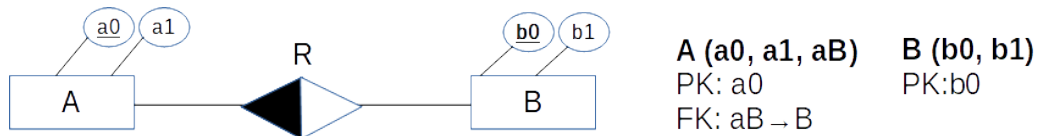
Si ambas entidades poseen cardinalidades (1,1) se puede propagar la clave de cualquiera de ellas a la tabla resultante de la otra.



Esta relación queda como una única tabla R con la clave principal a0 y clave alternativa b0. Su relación es tan fuerte que en el modelo relacional queda reflejado con una única tabla.

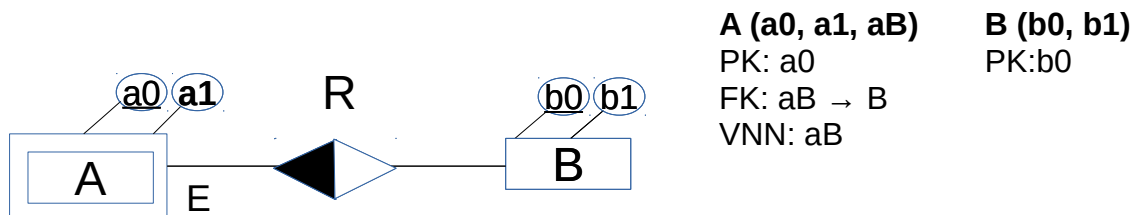
1.3.2.2. Relación 1:N

Se obtienen dos tablas. Una con los atributos de la entidad cuya cardinalidad máxima es 1, y otra con los atributos de la entidad cuya cardinalidad máxima es N, añadiéndole un atributo que corresponde con la clave principal de la primera tabla (clave ajena) y los atributos de la relación.

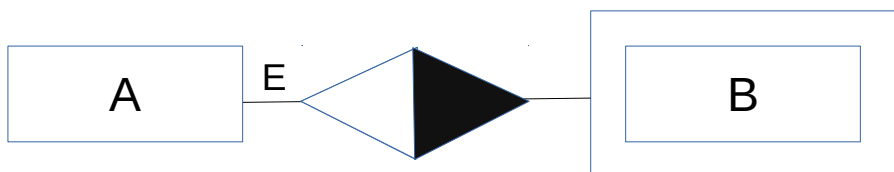


Si hay restricción de Existencia

Si tenemos restricción de existencia (siempre en el lado del muchos) sería igual que el anterior simplemente añadiendo valor no nulo.

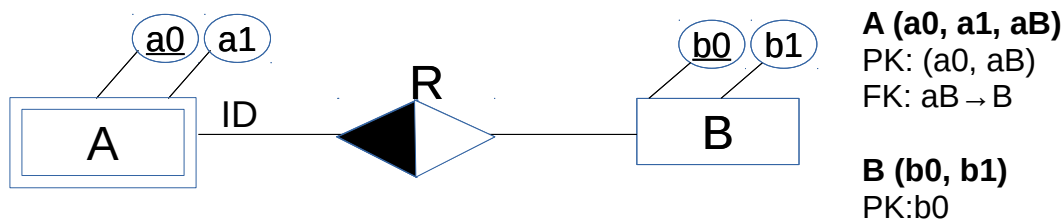


El siguiente caso sería una pérdida semántica porque no se puede representar de forma lógica:



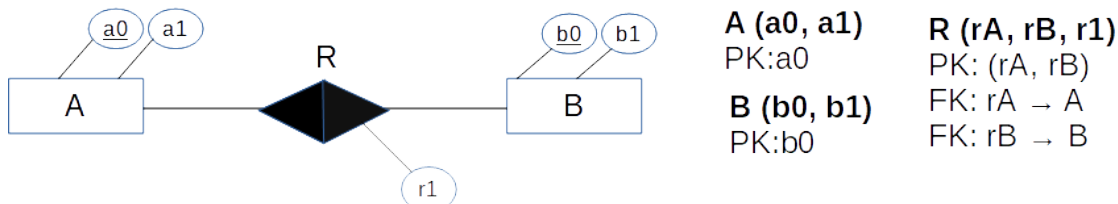
Restricción de identidad

Para el caso de una restricción de identidad, su representación en el modelo relacional sería el siguiente:



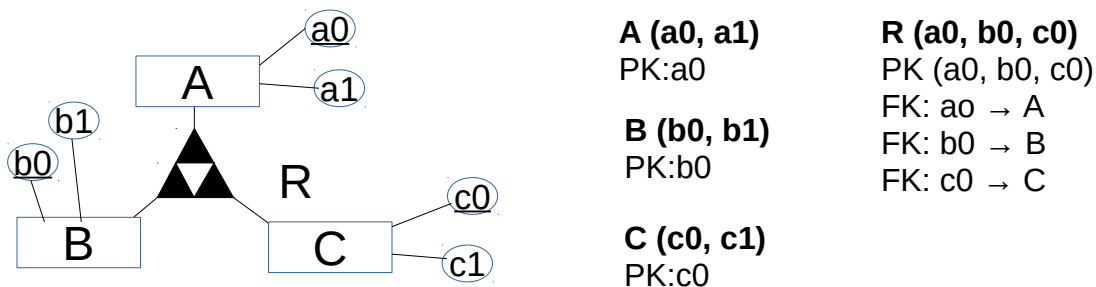
1.3.2.3. Relación N:N

Se obtienen tres tablas .Una con los atributos de la primera entidad. Otra con los de la segunda; y otra con los atributos de la relación más dos atributos que corresponden con las claves primarias de las tablas relacionadas que concatenadas actúan como clave principal, y que además, por separado actúan como claves ajenas de su tabla.

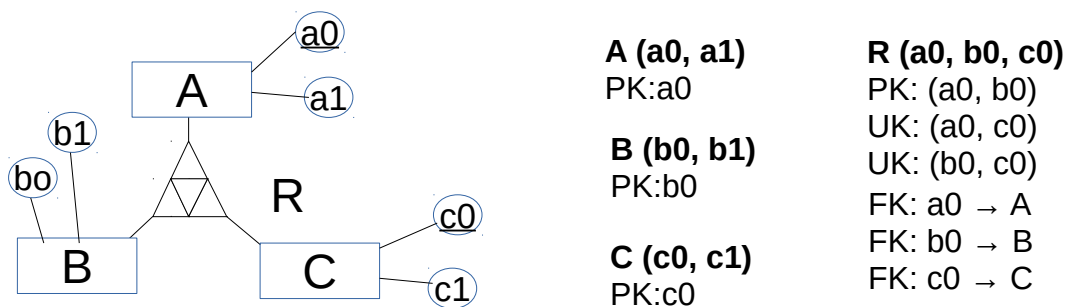


1.3.2.4. Relaciones Ternarias

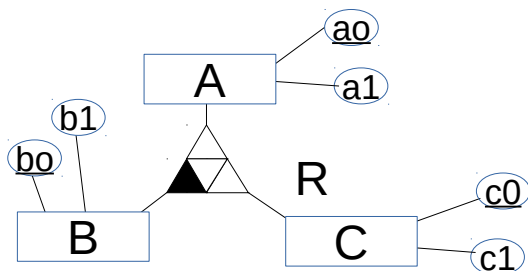
Relación Ternaria N:N:N: Se obtienen cuatro tablas .Tres con los atributos de cada Entidad. Y otra con los atributos de la relación más la propagación de las tres claves principales



Relación Ternaria 1:1:1



Relación Ternaria 1:1:N



A (a0, a1)
PK:a0

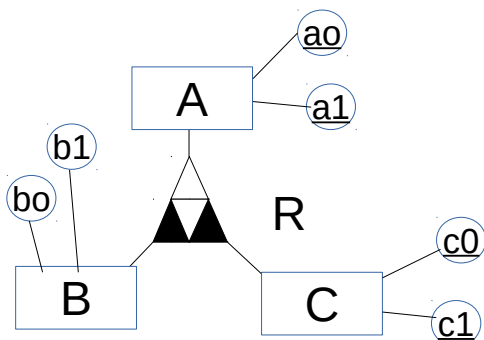
B (b0, b1)
PK:b0

C (c0, c1)
PK:c0

R (a0, b0, c0)
PK: (a0, b0)
UK: (b0, c0)

FK: a0 → A
FK: b0 → B
FK: c0 → C

Relación Ternaria 1:N:N



A (a0, a1)
PK:a0

B (b0, b1)
PK:b0

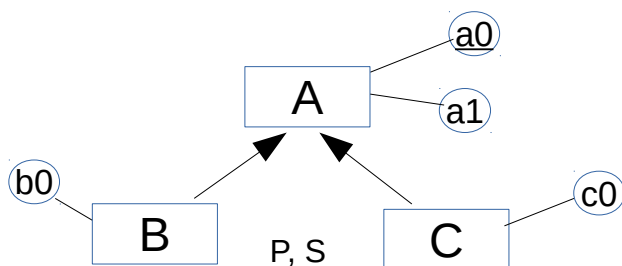
C (c0, c1)
PK:c0

R (a0, b0, c0)
PK: (b0, c0)
VNN: a0
FK: a0 → A
FK: b0 → B
FK: c0 → C

1.3.2.5.

Generalizaciones

En el modelo Relacional no se puede representar la generalización Total y Disjunta, sólo se puede representar en forma Parcial y Solapada, por lo que resto de casos son pérdidas semánticas.



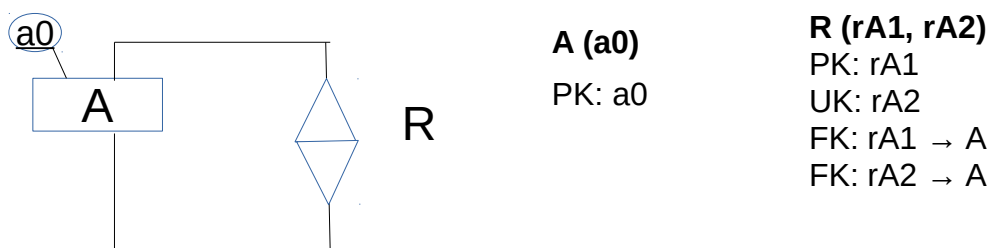
A (a0, a1)
PK: a0

B (b0, bA)
PK: bA
FK: bA → A

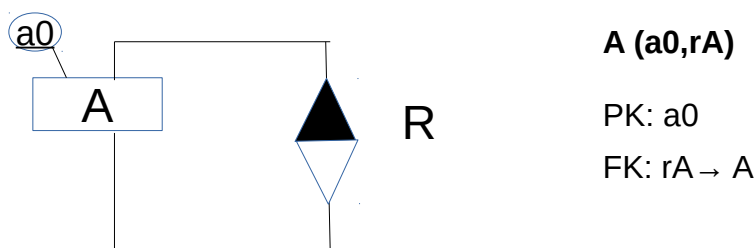
C (c0, cA)
PK: cA
FK: cA → A

1.3.2.6. Relaciones Reflexivas

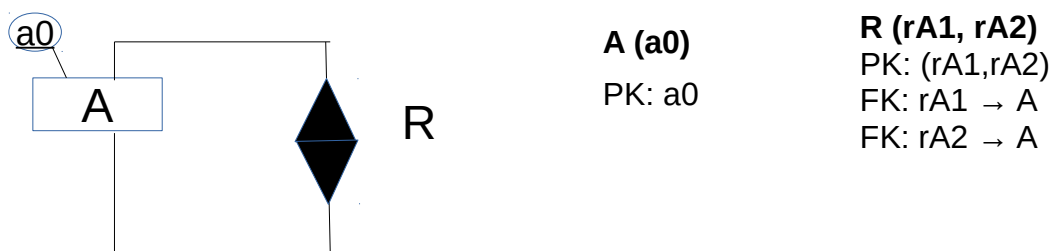
Las relaciones reflexivas se transforman siguiendo los mismos patrones que hemos visto hasta ahora, es decir si los ejemplos vistos la relación relaciona la entidad A con la entidad B, la entidad B pasa a comportarse como entidad A. Vemos unos ejemplos:



El atributo rA2 representa el mismo valor que rA1, lo único que es necesario duplicarlo para poder realizar la relación reflexiva sobre la propia entidad. Hacer notar, que tanto rA1 como rA2 son claves ajenas a la tabla A.

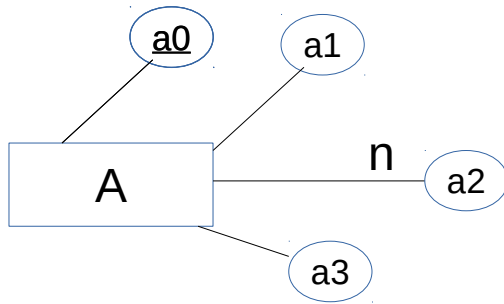


En la figura anterior se puede ver que esta relación quedaría como una única tabla.



1.3.2.7. Atributos multivaluados

Para estos atributos se generan tablas separadas que representen el atributo multivaluado incluido en la clave primaria de la entidad. A menos que el atributo multivaluado sea una clave alternativa de la entidad, la clave principal de la nueva tabla se forma con la combinación del atributo multivaluado y la clave principal de la entidad



A (a0, a1, a3)
PK: a0

A2 (a0, a2)
PK: (a0, a2)
FK: a0 → A

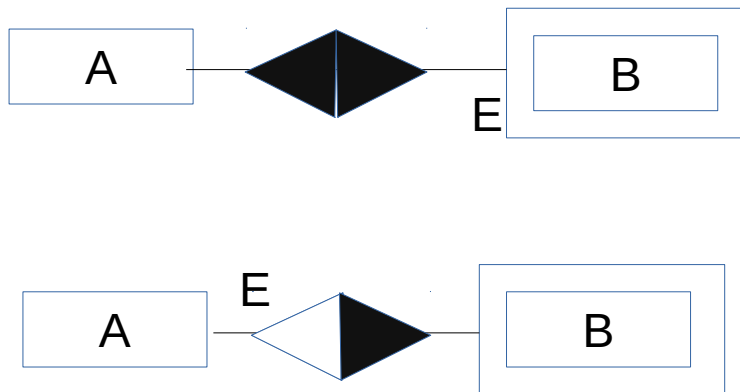
1.3.3. Pérdidas Semánticas

Es muy posible que lo que representamos en el modelo conceptual no sea representable en el modelo lógico o en nuestro caso el modelo relacional. A esto lo llamamos pérdidas semánticas.

Para solucionar las posibles pérdidas semánticas no podremos hacerlo a un nivel lógico, sino que nos esperaremos a la implementación para solucionarlo con disparadores o procedimientos almacenados, pero esto ya pertenece al lenguaje del Sistema Gestor de Base de Datos.

1.3.3.1. En Binarias

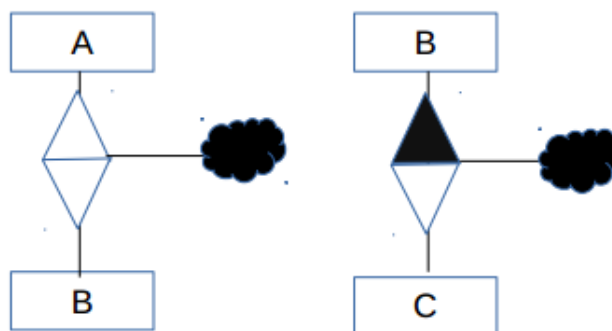
Las pérdidas semánticas que no encontramos en relaciones binarias son:



No se puede captar la restricción de existencia (E) de una entidad hacia una relación si la entidad y la relación no se encuentran en la misma tabla.

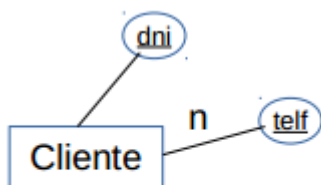
1.3.3.2. En Agregaciones

Las pérdidas semánticas que nos encontramos al convertir agregaciones son:



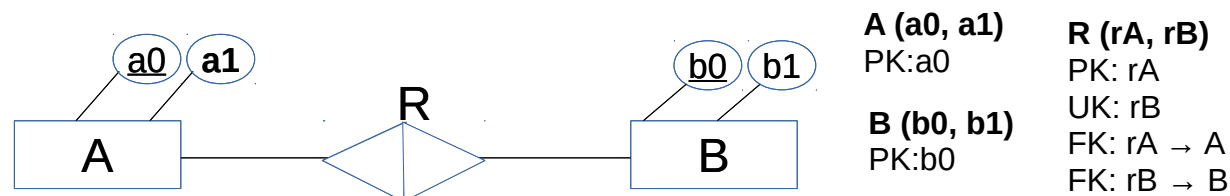
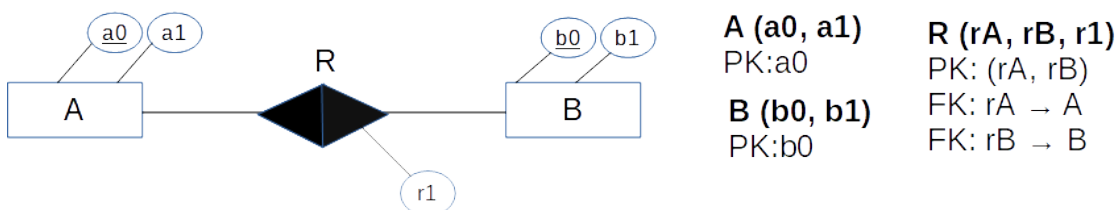
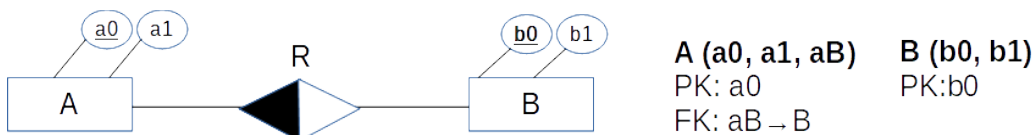
1.3.3.3. En atributos Multivaluados

Los atributos multivaluados también representan una pérdida semántica del modelo.

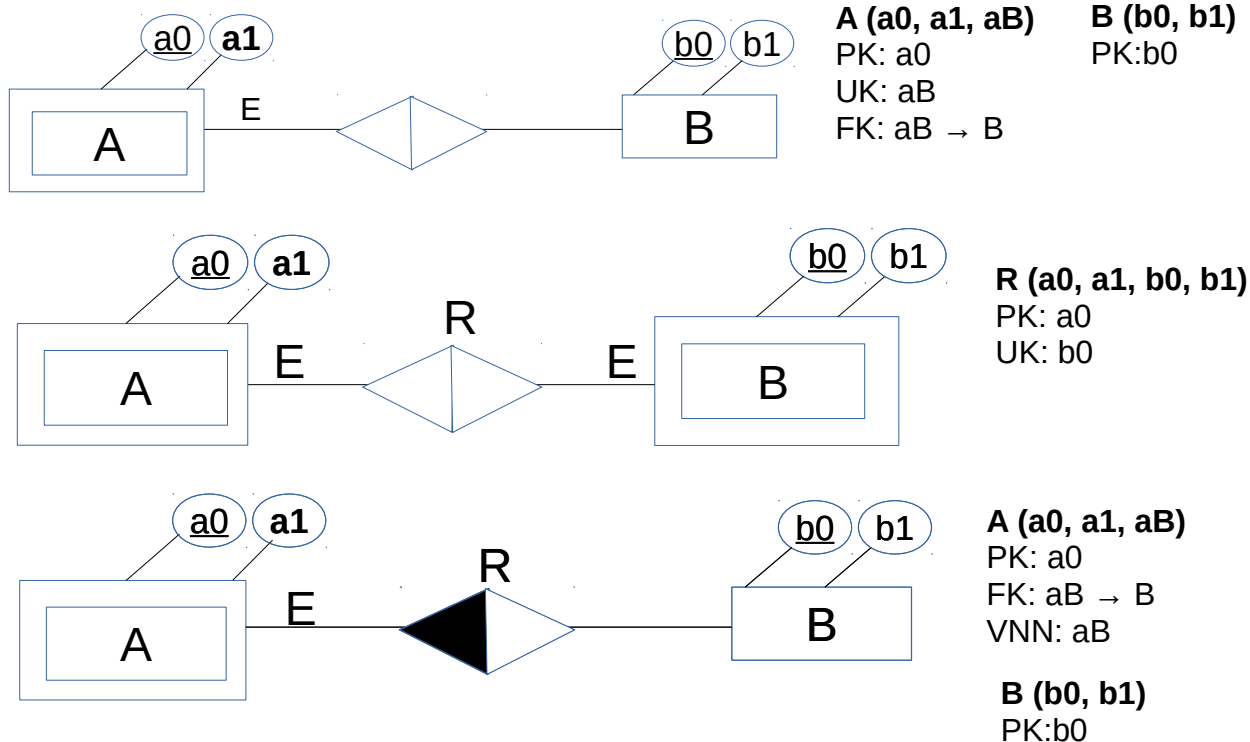


1.4. Tabla Resumen para la conversión de EER a MR

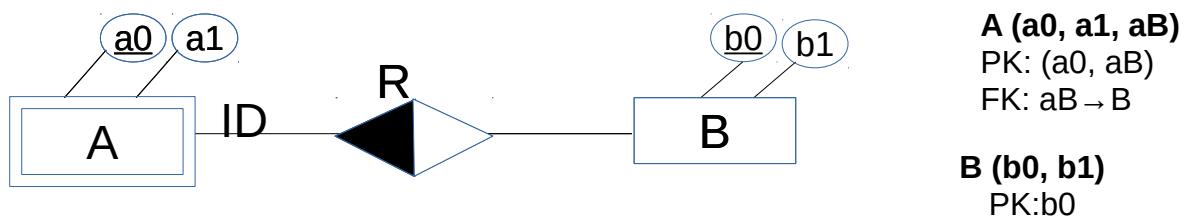
Binarias



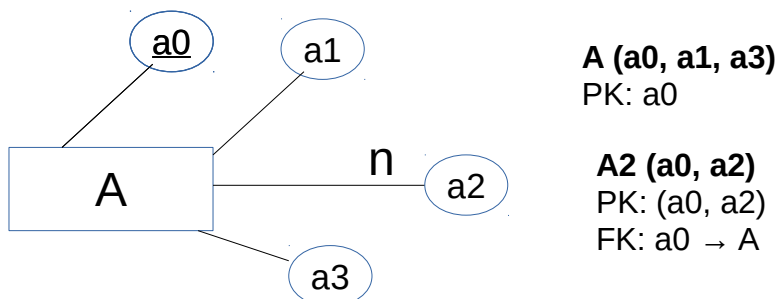
Con restricción de existencia (E)



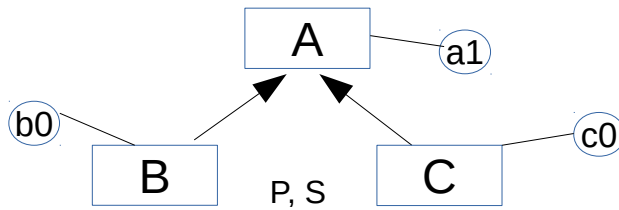
Con restricción de identidad



Multivaluados



Generalizaciones



A (a0, a1)

PK: a0

B (b0, bA)

PK: bA

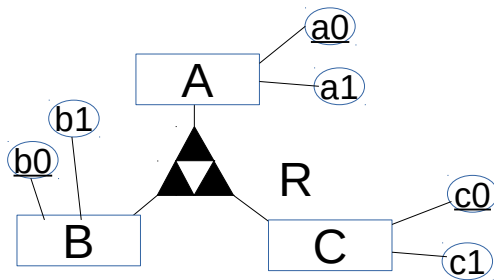
FK: bA → A

C (c0, cA)

PK: cA

FK: cA → A

Ternarias



A (a0, a1)

PK:a0

B (b0, b1)

PK:b0

R (a0, b0, c0)

PK (a0, b0, c0)

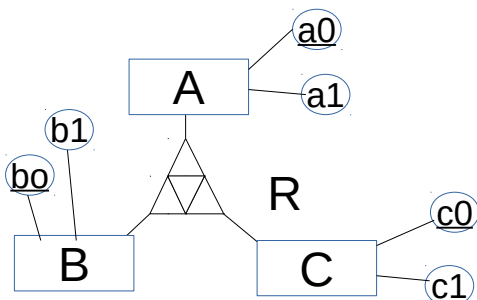
FK: a0 → A

FK: b0 → B

FK: c0 → C

C (c0, c1)

PK:c0



A (a0, a1)

PK:a0

B (b0, b1)

PK:b0

R (a0, b0, c0)

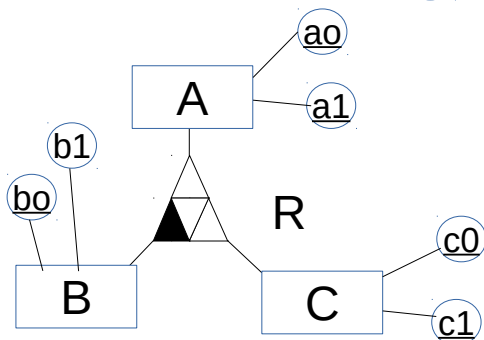
PK: (a0, b0)

UK: (a0, c0)

UK: (b0, c0)

C (c0, c1)

PK:c0



A (a0, a1)

PK:a0

B (b0, b1)

PK:b0

R (a0, b0, c0)

PK: (a0, b0)

UK: (b0, c0)

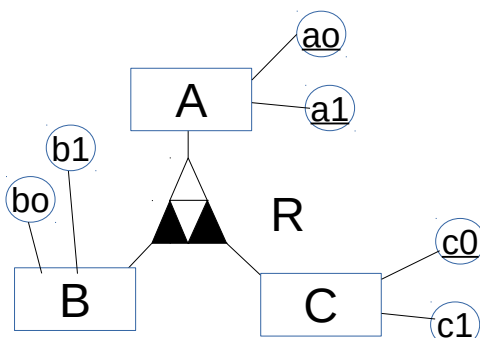
FK: a0 → A

FK: b0 → B

FK: c0 → C

C (c0, c1)

PK:c0



A (a0, a1)

PK:a0

B (b0, b1)

PK:b0

R (a0, b0, c0)

PK: (b0, c0)

VNN: a0

FK: a0 → A

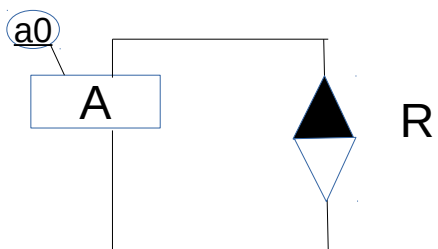
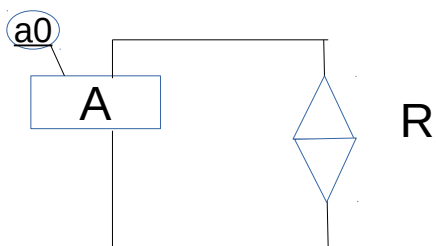
FK: b0 → B

FK: c0 → C

C (c0, c1)

PK:c0

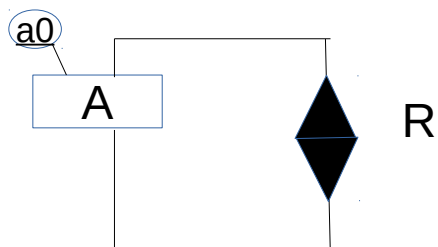
Relaciones reflexivas



A (a0,rA)

PK: a0

FK: rA \rightarrow A



Pérdidas semánticas

