



BASES DE DATOS

1r DAW - C.F.G.S.



UNIDAD 04

DISEÑO LÓGICO: CREACIÓN DEL ESQUEMA LÓGICO

ÍNDICE DE CONTENIDOS

1. INTRODUCCIÓN.....	3
2. REPRESENTACIÓN DE LAS ESTRUCTURAS.....	3
2.1 TRANSFORMACIÓN DE LAS ENTIDADES:.....	4
2.1.1 ENTIDADES FUERTES:.....	4
2.1.2 ENTIDADES DÉBILES:.....	5
2.1.3 ENTIDADES ESPECIALIZADAS:.....	7
2.2 TRANSFORMACIÓN DE LAS RELACIONES:.....	10
2.2.1 RELACIONES BINARIAS:.....	10
2.2.1.1 RELACIONES BINARIAS 1:1 SIN RESTRICCIÓN DE EXISTENCIA.....	10
2.2.1.2 RELACIONES BINARIAS 1:1 CON RESTRICCIÓN DE EXISTENCIA.....	11
2.2.1.3 RELACIÓN BINARIAS 1:1 CON DOS RESTRICCIONES DE EXISTENCIA.....	12
2.2.1.4 RELACIÓN BINARIAS 1:M SIN RESTRICCIONES DE EXISTENCIA.....	14
2.2.1.5 RELACIÓN BINARIA 1:M CON UNA RESTRICCIÓN DE EXISTENCIA SOBRE LA ENTIDAD DE CARDINALIDAD MÁXIMA 1.....	15
2.2.1.6 RELACIÓN BINARIA 1:M CON DOS RESTRICCIONES DE EXISTENCIA.....	17
2.2.1.7 RELACIÓN BINARIA M:M SIN RESTRICCIONES DE EXISTENCIA.....	17
2.2.2 RELACIONES REFLEXIVAS.....	18
2.2.2.1 RELACIÓN BINARIA REFLEXIVA 1:1 SIN RESTRICCIÓN DE EXISTENCIA.....	18
2.2.2.2 RELACIÓN BINARIA REFLEXIVA 1:1 CON RESTRICCIÓN DE EXISTENCIA.....	19
2.2.2.3 RELACIÓN REFLEXIVA 1:M SIN RESTRICCIÓN DE EXISTENCIA.....	20
2.2.2.4 RELACIÓN REFLEXIVA 1:M con una RESTRICCIÓN DE EXISTENCIA.....	20
2.2.2.5 RELACIÓN REFLEXIVA M:M SIN RESTRICCIONES DE EXISTENCIA.....	22
2.2.3 RELACIÓN TERNARIAS.....	23
2.2.3.1 RELACIÓN TERNARIAS M:M:M.....	25
2.2.3.2 RELACIÓN TERNARIAS 1:M:M.....	26
2.2.3.3 RELACIÓN TERNARIAS 1:1:M.....	27
2.2.3.4 RELACIÓN TERNARIAS 1:1:1.....	28
2.3 ATRIBUTOS EN LAS RELACIONES.....	29
2.4 TRANSFORMACIÓN DE LA AGREGACIÓN.....	35
2.5 RESUMEN.....	39

1. INTRODUCCIÓN.

El diseño lógico consiste en la transformación del esquema conceptual, que se encuentra descrito con un cierto modelo de datos, en estructuras y transacciones descritas en términos del modelo de datos en el cual se base el sistema de gestión de bases de datos que se vaya a utilizar. En este tema se presenta este diseño lógico teniendo en cuenta que el modelo de datos elegido es el modelo relacional; así, se deberá realizar este proceso para obtener finalmente un esquema relacional y un conjunto de transacciones sobre las relaciones de este esquema. Este proceso se dividirá en dos fases:

1) Transformación de los aspectos estáticos del esquema conceptual: para ello se verá cómo es posible transformar cada una de las estructuras de un diagrama entidad-relación, modelo utilizado en el diseño conceptual, en relaciones. Aquí el término *relación* hace referencia a la estructura de datos del modelo relacional (en inglés *relation*) y no hay que confundirlo con la relación del modelo entidad-relación (en inglés *relationship*).

Esta transformación de los aspectos estáticos del esquema conceptual dará lugar a un primer **esquema lógico** (Esquema relacional). Además, en la segunda parte de este tema se estudiará la **teoría de la normalización**, que permite refinar este primer esquema lógico obteniendo un esquema adecuado desde el punto de vista de su manipulación.

2) Transformación de los aspectos dinámicos del esquema conceptual: para ello se verá cómo se obtienen transacciones sobre las relaciones del esquema relacional a partir del análisis de transacciones realizado en la fase de diseño conceptual.

2. REPRESENTACIÓN DE LAS ESTRUCTURAS.

El proceso de obtención de un esquema relacional que represente adecuadamente todos los aspectos estáticos expresados en el esquema conceptual (que están descritos en el diagrama entidad-relación y en el conjunto de restricciones de integridad añadidas) consiste en **aplicar un conjunto de reglas** para transformar el diagrama entidad-relación en un **esquema lógico**, que constará de un conjunto de relaciones que lo representen adecuadamente. En esta transformación también tienen que ser tratadas las posibles restricciones del esquema conceptual, traduciéndolas a expresiones equivalentes del SQL o, en algunos casos, muy pocos, integrándolas en la propia definición de las relaciones.

El objetivo de este apartado es presentar estas transformaciones. Para ello se supondrá que el sistema de gestión de bases de datos relacionales soporta, para cada relación, la definición de clave primaria, clave ajena, restricción de unicidad y restricción de valor no nulo sobre atributos.

En algunos casos puede suceder que haya varios esquemas relacionales posibles para un mismo esquema conceptual; el criterio de elección que se aplicará cuando esto suceda es el siguiente: "Elige el esquema con menos restricciones de integridad añadidas. Ante igualdad de restricciones, elige el esquema con menos relaciones resultantes". Este criterio se justifica por el hecho de que las restricciones de integridad suponen, usualmente, controles costosos en tiempo; por otra parte, cuantas menos relaciones tenga el esquema más eficientes serán las operaciones de consulta. Sólo en algunos casos muy concretos puede no seguirse este criterio.

Para estudiar estas transformaciones, vamos a realizar un barrido por los posibles objetos y estructuras de un diagrama entidad-relación, presentándose en cada caso el conjunto de relaciones equivalentes.

2.1 TRANSFORMACIÓN DE LAS ENTIDADES:

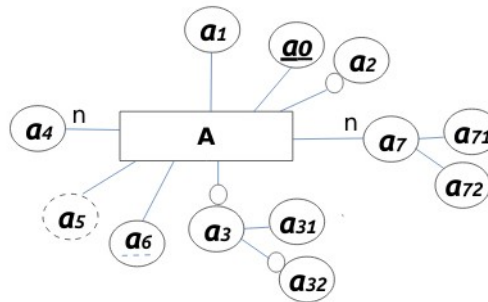
En un diagrama entidad-relación, se pueden distinguir tres clases de entidades:

- Entidades fuertes (ni débiles ni especializadas),
- Entidades débiles, y
- Entidades especializadas (o generalizadas).

A continuación se muestra cómo se transforma cada uno de ellos.

2.1.1 ENTIDADES FUERTES:

Veamos una entidad fuerte con su conjunto de atributos como en la figura.



La relación equivalente es la siguiente:

A (**a0**: dom_a0, **a1**: dom_a1, **a2**: dom_a2, **a3.a31**: dom_a31, **a3.a32**: dom_a32, {**a4**: dom_a4}, **a5**: dom_a5, **a6**: dom_a6, {**a7.a71**: dom_a71, **a7.a72**: dom_a72})

Clave Primaria: {**a0**}

Valor No Nulo: {**a2**}

Valor No Nulo: {**a3.a32**}

Único: {**a6**}

Donde:

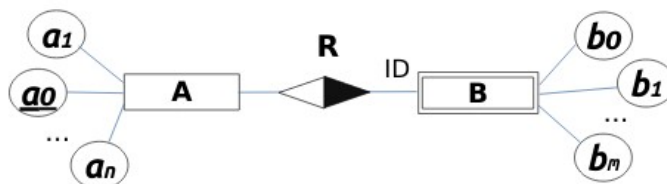
- Los dominios de los atributos de la relación se obtienen del anexo del diagrama entidad-relación, donde se asocian dominios a los atributos.
- El atributo identificador (en el ejemplo **a0**) se convierte en la clave primaria de la relación. Si la entidad tuviera un identificador formado por más de un atributo, el conjunto de todos ellos formaría la clave primaria de la relación.
- Las restricciones de valor no nulo (p.e. **a2**) y las restricciones de unicidad (p.e. **a6**) se representan sin problemas como se muestra en el ejemplo.
- Los atributos multivaluados (p.e. **a4**) se incluyen en la relación destacados, cada uno de ellos, entre llaves (Con esta notación se destaca el hecho de que ese atributo toma para cada tupla múltiples valores.).

Los problemas que puedan derivarse de la presencia de estos atributos se resolverán en la aplicación de las reglas de normalización, las cuales trataremos más adelante.

- Los atributos compuestos (p.e. **a3**) se representan descomponiéndolos en sus atributos componentes. Esta descomposición podría posponerse hasta la fase de normalización, pero el realizarla directamente permite especificar algunas restricciones de valor no nulo sobre los atributos componentes. Por otra parte, si no hay problemas de ambigüedad con los nombres de los atributos, podrá utilizarse como nombre el descriptor del campo (**a31**, **a32**,...). Obsérvese cómo se ha representado el atributo **a7** que es un atributo compuesto multivaluado.
- Los atributos derivados (calculados) (p.e. **a5**) se incluyen en la relación como un atributo más, debiendo especificarse la fórmula con la que se derivará su valor como si se tratara de una restricción más.

2.1.2 ENTIDADES DÉBILES:

La transformación de una entidad débil es análoga a la de una entidad que no sea débil; la única diferencia es que es necesario incorporar, como atributos de la entidad débil, la clave primaria de la entidad fuerte. Además, la clave primaria de la entidad débil estará formada por la clave primaria de la entidad fuerte más un conjunto de atributos propio de la entidad débil (**discriminante**). Para ilustrar esta transformación, supóngase el siguiente diagrama entidad-relación:



A (a0: dom_a0, a1: dom_a1,..., an: dom_an)

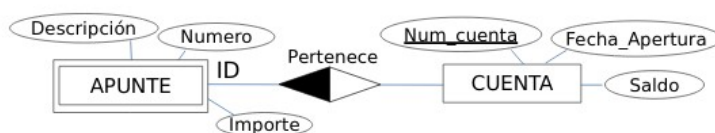
Clave Primaria: {a0}

B (b0: dom_b0, b1: dom_b1,..., bm: dom_bm, a0: dom_a0)

Clave Primaria: {a0, b0}

Clave Ajena: {a0} hace referencia a A

EJEMPLO:



CUENTA (Num_cuenta: dom_C, Fecha_Apertura: dom_Fecha, Saldo: dom_Saldo)

Clave Primaria: {Num_cuenta}

APUNTE (Num_cuenta: dom_C, Numero: dom_Num, Descripcion: dom_Des, Importe: dom_I)

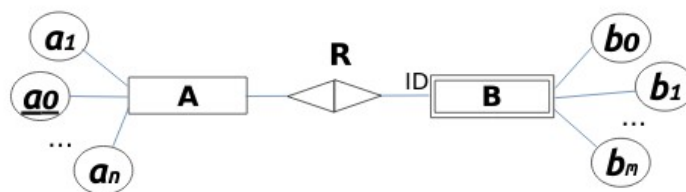
Clave Primaria: {Num_cuenta, Numero}

Clave Ajena: {Num_cuenta} hace referencia a CUENTA

CUENTA			APUNTE			
Num_cuenta	Fecha_Apertura	Saldo	Num_cuenta	Numero	Descripcion	Importe
{Clave Primaria}			{Clave Primaria}			
2600548212521	20/06/19	2.066,00 €	2600548212521	1	~~~~~	2000
			2600548212521	2	~~~~~	66

En este ejemplo el atributo Saldo de la entidad CUENTA podríamos considerarlo como un atributo calculado que cambia su valor cada vez que se modifica un apunte...

Existe otro diagrama en el que puede aparecer una restricción de identificación; es el siguiente:



cuyo esquema relacional equivalente es:

A (a0: dom_a0, a1: dom_a1,..., an: dom_an)

Clave Primaria: {a0}

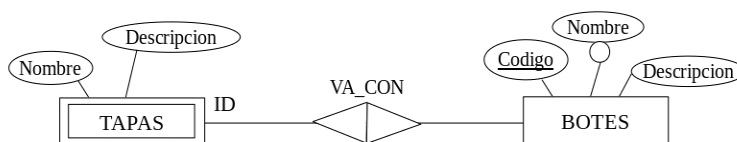
B (b0: dom_b0, b1: dom_b1,..., bm: dom_bm, a0: dom_a0)

Clave Primaria: {a0}

Clave Ajena: {a0} hace referencia a A

Hay que darse cuenta de que con estas transformaciones no sólo se representan las dos entidades, sino que también queda representada la relación R.

EJEMPLO:



BOTES (Codigo: dom_Cod, Nombre: dom_nom, Descripcion: dom_desc)

Clave Primaria: {Codigo}

TAPAS (Codigo: dom_Cod, Nombre: dom_nom, Descripcion: dom_desc)

Clave Primaria: {Codigo}

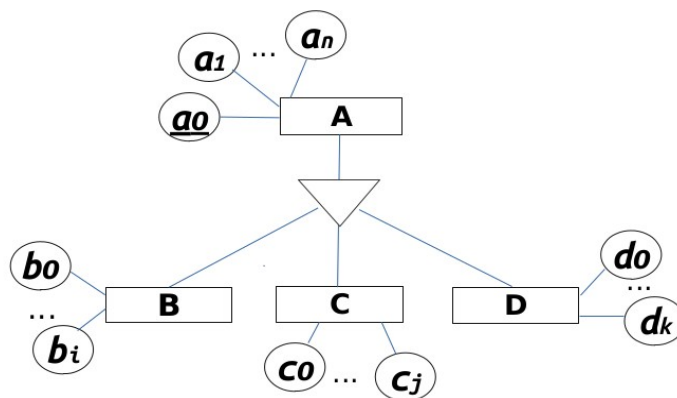
Clave Ajena: {Codigo} hace referencia a BOTES

TAPAS			BOTES		
Codigo	Nombre	Descripcion	Codigo	Nombre	Descripcion
{Clave Primaria}			{Clave Primaria}		
BTR12	Redonda-12	~~~~~	BTR12	Bote-Alto-12	~~~~~
BTC6	Exagonal-6	~~~~~	BTC6	Bote-Medio6-2	
			{Clave Ajena}		

Como se puede observar, al ser los códigos de los botes claves primarias en las dos relaciones, la relación 1:1 se establece perfectamente así como las tapas quedan perfectamente identificadas por sus botes.

2.1.3 ENTIDADES ESPECIALIZADAS:

Sólo cuando la especialización es Parcial y Solapada existe una transformación en relaciones totalmente adecuada; en los demás casos es necesario incluir ciertas restricciones de integridad que permitan la definición exacta de estos objetos. La transformación consiste en definir una relación para cada entidad especializada que incluye los atributos propios y también la clave primaria de la relación que representa a la entidad general, que pasa a ser también la clave primaria de la relación. Sea, por ejemplo, el siguiente diagrama entidad-relación:



- A** (a_0 : dom_a0,
a1: dom_a1,..., an: dom_an)
Clave Primaria: {a0}
- B** (a_0 : dom_a0, b1: dom_b1,..., bi: dom_bi)
Clave Primaria: {a0}
Clave Ajena: {a0} hace referencia a A
- C** (a_0 : dom_a0, c1: dom_c1,..., cj: dom_cj)
Clave Primaria: {a0}
Clave Ajena: {a0} hace referencia a A
- D** (a_0 : dom_a0, d1: dom_d1,..., dk: dom_dk)
Clave Primaria: {a0}
Clave Ajena: {a0} hace referencia a A

Obsérvese que la clave primaria de las relaciones donde se representan las entidades especializadas se define también como clave ajena a la relación donde está representada la entidad general; de esta forma se expresa la restricción de que toda ocurrencia de cualquier entidad especializada corresponde a una ocurrencia de la entidad general.

En caso de que la generalización sea de otro tipo, el esquema relacional que se debe definir es el mismo, pero es necesaria la inclusión de algunas restricciones de integridad como se muestra a continuación:

- **Total** y Solapada: En este caso el conjunto de relaciones anterior no representa exactamente el objeto general ya que queda por expresar el hecho de que toda ocurrencia de la entidad general A tiene que estar asociada con al menos

una ocurrencia de alguna entidad especializada. Es necesaria la definición de una restricción de integridad como se muestra:

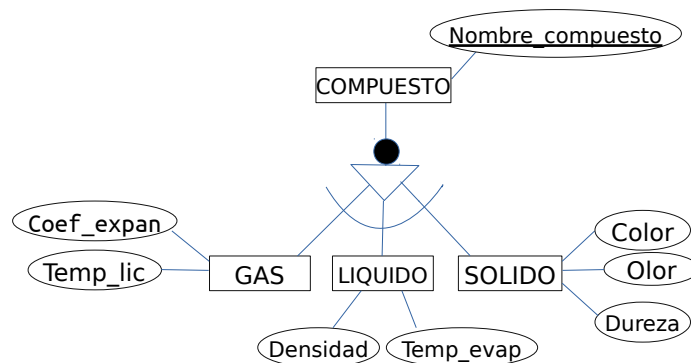
```
CREATE ASSERTION Total CHECK
  NOT EXISTS (SELECT Ax.a0 FROM A Ax
    WHERE Ax.a0 NOT IN (SELECT Bx.a0 FROM B Bx UNION
      SELECT Cx.a0 FROM C Cx UNION
      SELECT Dx.a0 FROM D Dx))
```

- Parcial y **Disjunta**: Para este tipo de especialización, el conjunto de relaciones anterior no expresa la restricción de que las entidades especializadas son disjuntas, esto es, el hecho de que cada ocurrencia de la entidad general sólo puede estar asociada con una ocurrencia de una entidad especializada. La expresión en SQL que representa esta restricción es la siguiente:

```
CREATE ASSERTION Disjunta CHECK
  NOT EXISTS ( SELECT * FROM B Bx, C Cx, D Dx
    WHERE (Dx.a0 = Bx.a0 OR Dx.a0 = Cx.a0 OR Bx.a0 = Cx.a0))
```

- Total y Disjunta: En este caso es necesaria la inclusión de las dos restricciones anteriores para obtener un esquema relacional que represente exactamente este tipo de objeto general.

EJEMPLO:



COMPUESTO (**Nombre_compuesto**)

Clave Primaria: {Nombre_compuesto}

GAS (**Nombre_comp_gas**, Coef_expan, Temp_lic)

Clave Primaria: {Nombre_comp_gas}

Clave Ajena: {Nombre_comp_gas} hace referencia a COMPUESTO

LIQUIDO (**Nombre_comp_liq**, Densidad, Temp_evap)

Clave Primaria: {Nombre_compuesto_liq}

Clave Ajena: {Nombre_comp_liq} hace referencia a COMPUESTO

SOLIDO (**Nombre_comp_sol**, Color, Olor, Dureza)

Clave Primaria: {Nombre_compuesto_sol}

Clave Ajena: {Nombre_comp_sol} hace referencia a COMPUESTO


```
CREATE ASSERTION Total CHECK
    NOT EXISTS (SELECT Nombre_compuesto FROM COMPUESTO Comp
        WHERE Comp.Nombre_compuesto NOT IN
            (SELECT Gs.Nombre_comp_gas FROM GAS Gs
                UNION
                SELECT Lq.Nombre_comp_liq FROM LIQUIDO Lq
                UNION
                SELECT Sd.Nombre_comp_sol FROM SOLIDO Sd))
```

SELECT Nombre_comp_gas FROM GAS Gs → Todas las CP de Gaseoso
 SELECT Nombre_comp_liq FROM LIQUIDO Lq → Todas las CP de Líquido
 SELECT Nombre_comp_sol FROM SOLIDO Sd → Todas las CP de Sólido

```
CREATE ASSERTION Total CHECK
    NOT EXISTS (SELECT Nombre_compuesto FROM COMPUESTO Comp
        WHERE Comp.Nombre_compuesto NOT IN
            (Todas los Nombres de los compuestos en GAS, LIQUIDO y SOLIDO))
```

En definitiva con esta restricción se vigila que no haya (No existirá) ningún nombre de Compuesto que NO esté en GAS o en LIQUIDO o en SOLIDO.

```
CREATE ASSERTION Disjunta CHECK
    NOT EXISTS ( SELECT * FROM GAS Gs, LIQUIDO Lq, SOLIDO Sd
        WHERE (Sd.Nombre_comp_sol = Lq.Nombre_comp_liq
            OR
            Lq.Nombre_comp_liq = Gs.Nombre_comp_gas
            OR
            Gs.Nombre_comp_gas = Sd.Nombre_comp_sol))
```

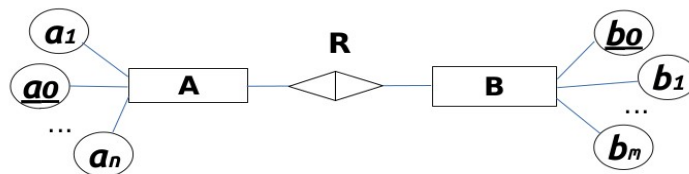
Con esta restricción se vigila que no exista un compuesto que sea solido y líquido a la vez o líquido y gas a la vez o gas y sólido al tiempo.

2.2 TRANSFORMACIÓN DE LAS RELACIONES:

Vamos a considerar diecisiete casos que abarcan relaciones binarias, reflexivas y ternarias; después de presentar todos estos casos se estudiará la presencia de atributos en la relación.

2.2.1 RELACIONES BINARIAS:

2.2.1.1 RELACIONES BINARIAS 1:1 SIN RESTRICCIÓN DE EXISTENCIA



Las relaciones equivalentes a esta estructura son:

Esquema 1

- A** (a0: dom_a0, a1: dom_a1,..., an: dom_an)
Clave Primaria: {a0}
- B** (b0: dom_b0, b1: dom_b1,..., bm: dom_bm, a0: dom_a0)
Clave Primaria: {b0}
Único: {a0}
Clave Ajena: {a0} hace referencia a A

NOTA: Sería más adecuado que la relación B se llamara B-R ya que en ella se representan la entidad B y la relación R, sin embargo por comodidad en todo el documento se ha elegido exclusivamente el nombre de la entidad.

Como puede observarse, la representación de la relación R se realiza mediante la inclusión en la relación B de una clave ajena a0 que hace referencia a la relación A y que además se define también con restricción de unicidad para representar correctamente que la cardinalidad máxima de A es 1.

Este esquema relacional no es el único posible, también podría elegirse el siguiente esquema, en el que la relación R se ha representado junto con A:

Esquema 2

- A** (a0: dom_a0, a1: dom_a1,..., an: dom_an, b0: dom_b0)
Clave Primaria: {a0}
Único: {b0}
Clave Ajena: {b0} hace referencia a B
- B** (b0: dom_b0, b1: dom_b1,..., bm: dom_bm)
Clave Primaria: {b0}

Además de estos dos esquemas, también podrían diseñarse los siguientes que representan correctamente la realidad modelada en el anterior diagrama entidad-relación pero que son menos adecuados por constar de una relación más.

En los demás casos que se van a estudiar, no se mostrarán todos los esquemas posibles sino tan sólo el que se considere mejor o al menos tan bueno como todos los esquemas posibles.

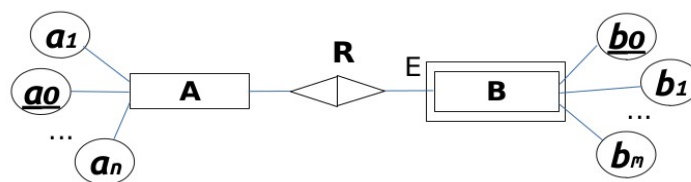
Esquema 3

- A** (a0: dom_a0, a1: dom_a1,..., an: dom_an)
Clave Primaria: {a0}
- B** (b0: dom_b0, b1: dom_b1,..., bm: dom_bm)
Clave Primaria: {b0}
- R** (b0: dom_b0, a0: dom_a0)
Clave Primaria: {b0}
Único: {a0}
Valor No Nulo: {a0}
Clave Ajena: {a0} hace referencia a A
Clave Ajena: {b0} hace referencia a B

Esquema 4

- A** (a0: dom_a0, a1: dom_a1,..., an: dom_an)
Clave Primaria: {a0}
- B** (b0: dom_b0, b1: dom_b1,..., bm: dom_bm)
Clave Primaria: {b0}
- R** (b0: dom_b0, a0: dom_a0)
Clave Primaria: {a0}
Único: {b0}
Valor No Nulo: {b0}
Clave Ajena: {a0} hace referencia a A
Clave Ajena: {b0} hace referencia a B

2.2.1.2 RELACIONES BINARIAS 1:1 CON RESTRICCIÓN DE EXISTENCIA



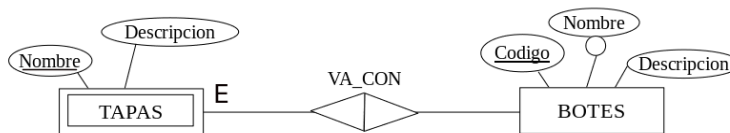
Las relaciones equivalentes a esta estructura son:

Esquema 1

- A** (a0: dom_a0, a1: dom_a1,..., an: dom_an)
Clave Primaria: {a0}
- B** (b0: dom_b0, b1: dom_b1,..., bm: dom_bm, a0: dom_a0)
Clave Primaria: {b0}
Único: {a0}
Valor No Nulo: {a0}
Clave Ajena: {a0} hace referencia a A

En este caso, como la entidad B está obligada a relacionarse con la entidad A a través de la relación R, es posible representar esta relación mediante la inclusión, en la relación que representa a la entidad B de una clave ajena con restricción de valor no nulo para representar la restricción de existencia y con restricción de unicidad para representar correctamente la cardinalidad máxima de A.

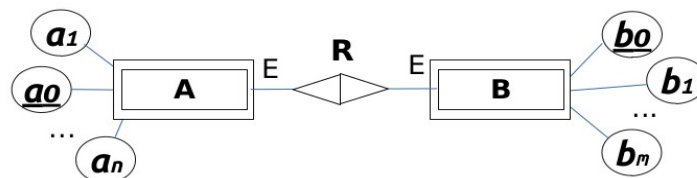
EJEMPLO:



BOTES (**Código**, Nombre, Descripción)
Clave Primaria: {Código}

TAPAS (**Nombre**, Descripción, **Código**)
Clave Primaria: {Nombre}
Unico: {Código}
Valor No Nulo: {Código}
Clave Ajena: {Código} hace referencia a BOTES

2.2.1.3 RELACIÓN BINARIAS 1:1 CON DOS RESTRICCIONES DE EXISTENCIA



El esquema relacional equivalente a esta estructura contiene una única relación que incluye toda la información representada por las dos entidades y por la relación R, Este es probablemente el único caso en el que no se define una relación por cada entidad del diagrama.

A-B (a0: dom_a0, a1: dom_a1,..., an: dom_an, b0: dom_b0, b1: dom_b1,..., bm: dom_bm)
Clave Primaria: {a0}
Único: {b0}
Valor No Nulo: {b0}

Como se puede observar, la representación de esta estructura da lugar a una única relación, no siendo posible representar independientemente las dos entidades A y B sin tener que añadir restricciones de integridad. Sin embargo, esta solución complica la manipulación de los objetos representados por la entidad B (al no tener una relación propia) así que en algunos casos será preferible el siguiente esquema aunque tenga más relaciones y también restricciones de integridad:

A (a0: dom_a0, a1: dom_a1,..., an: dom_an, b0: dom_b0)

Clave Primaria: {a0}

Único: {b0}

Valor No Nulo: {b0}

Clave Ajena: {b0} hace referencia a B

B (b0: dom_b0, b1: dom_b1,..., bm: dom_bm)

Clave Primaria: {b0}

En este caso la elección de un esquema u otro dependerá de cada caso concreto.

La restricción de existencia de la entidad B respecto a la relación R podría representarse con la siguiente expresión.

```
CREATE ASSERTION Rest_existencia CHECK
  NOT EXISTS ( SELECT * FROM B Bx
               WHERE NOT EXISTS (SELECT * FROM A Ax
                                WHERE Bx.b0=Ax.b0))
```

EJEMPLO:



BOTES (Codigo, Nombre, Descripcion)

Clave Primaria: {Codigo}

TAPAS (Nombre, Descripcion, Codigo)

Clave Primaria: {Nombre}

Unico: {Codigo}

Valor No Nulo: {Codigo}

Clave Ajena: {Codigo} hace referencia a BOTES

TAPAS			BOTES		
Nombre	Descripcion	Codigo	Codigo	Nombre	Descripcion
{Clave Primaria}		{Único, VNN}	{Clave Primaria}		
Redonda-12	~~~~~	BTR12	BTR12	Bote-Alto-12	~~~~~
Exagonal-6	~~~~~	BTC6	BTC6	Bote-Medio6-2	~~~~~
		{Clave Ajena}			
Redonda-3x8	~~~~~	¿BT038?	BT038	Bote-Medio3x8	~~~~~

En este ejemplo vemos que una de la restricciones de existencia, la que recae sobre TAPAS, se resuelve con definir como valor no nulo la clave ajena que hace referencia a BOTES.

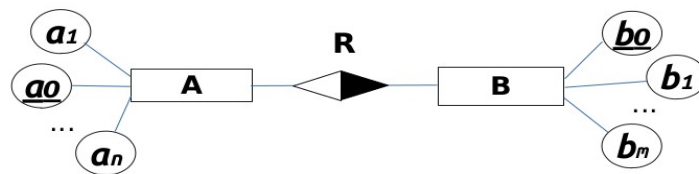
Para resolver la otra restricción de existencia, la que cae sobre BOTES y obliga a que al dar de alta un bote debe existir una tapa asociada, se debe definir la

expresión presentada anteriormente. Observa en la imagen anterior como a la hora de insertar el bote de código BT038 es preciso “vigilar” en una relación distinta que realmente ese bote tenga tapa.

```
CREATE ASSERTION Rest_existencia CHECK
  NOT EXISTS ( SELECT * FROM BOTES Bx
              WHERE NOT EXISTS (SELECT * FROM TAPAS Tx
                                WHERE Bx.Codigo=Tx.Codigo))
```

Aquí se da una situación de interbloqueo que debe ser resuelta programando una transacción que permita insertar un BOTE y su TAPA a la vez.

2.2.1.4 RELACIÓN BINARIAS 1:M SIN RESTRICCIONES DE EXISTENCIA



Las relaciones equivalentes a esta estructura son:

- A** (a0: dom_a0, a1: dom_a1,..., an: dom_an)
Clave Primaria: {a0}
- B** (b0: dom_b0, b1: dom_b1,..., bm: dom_bm, a0: dom_a0)
Clave Primaria: {b0}
Clave Ajena: {a0} hace referencia a A

En este caso se puede observar que debido a la cardinalidad de la relación binaria R es posible representar ésta como una clave ajena en la relación que representa la entidad con cardinalidad máxima 1 (la entidad B). Nótese que la posibilidad de que haya ocurrencias de la entidad B que no tomen parte en la relación R está bien trasladada al esquema relacional, ya que es posible que en tuplas de la relación B el atributo correspondiente a la clave ajena tenga valor nulo.

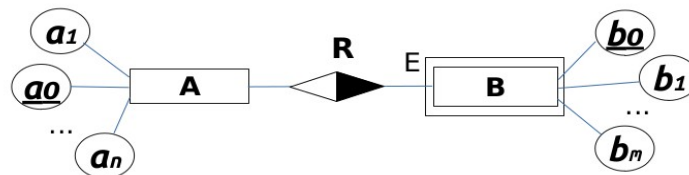
EJEMPLO:



MEDICO			PACIENTE			
Ncolegiado	Nombre	Descripcion	Codigo	NSS	Nombre	Doctor
{Clave Primaria}	{Valor No Nulo}		{Clave Primaria}	{VNN, Único}	{Valor No Nulo}	
4600252125	ANA GIL	~~~~~	P0002	~~~~~	Jose Alberto Ruiz	NULL
4680025003	PEP SANZ	~~~~~	P1112	~~~~~	Ana M.ª Soler	NULL
4620360001	EVA BAS	~~~~~	P8886	~~~~~	Carlos Andrés Pau	4680025003
			P7773	~~~~~	Jose M.ª Gomez	4680025003
			P3333	~~~~~	M.ª Carmen Bens	4680025003
			P5550	~~~~~	Julio José Oliver	4620360001

En el ejemplo queda claro cómo se establece la relación mediante la clave ajena “Doctor”. Obviamente esta clave ajena y la clave primaria “Ncolegiado” deberán compartir el mismo dominio.

2.2.1.5 RELACIÓN BINARIA 1:M CON UNA RESTRICCIÓN DE EXISTENCIA SOBRE LA ENTIDAD DE CARDINALIDAD MÁXIMA 1



Las relaciones equivalentes a esta estructura son:

- A** (a0: dom_a0, a1: dom_a1,..., an: dom_an)
Clave Primaria: {a0}
- B** (b0: dom_b0, b1: dom_b1,..., bm: dom_bm, a0: dom_a0)
Clave Primaria: {b0}
Clave Ajena: {a0} hace referencia a A
Valor No Nulo: {a0}

En este caso, la restricción de existencia de la entidad B se representa por la definición de una restricción de valor no nulo sobre la clave ajena de la relación B que representa la relación binaria R.

EJEMPLO:



MEDICO			PACIENTE			
Ncolegiado	Nombre	Descripcion	Codigo	NSS	Nombre	Doctor
{Clave Primaria}	{Valor No Nulo}		{Clave Primaria}	{VNN, Único}	{Valor No Nulo}	{Valor No Nulo}
4600252125	ANA GIL	~~~~~	P0002	~~~~~	Jose Alberto Ruiz	NULL
4680025003	PEP SANZ	~~~~~	P1112	~~~~~	Ana M.ª Soler	NULL
4620360001	EVA BAS	~~~~~	P8886	~~~~~	Carlos Andrés Pau	4680025003
			P7773	~~~~~	Jose M.ª Gomez	4680025003
			P3333	~~~~~	M.ª Carmen Bens	4680025003
			P5550	~~~~~	Julio José Oliver	4620360001

Claramente vemos que ahora no podemos tener a un paciente sin doctor asociado. Al querer insertar un paciente nuevo, el valor para el atributo “Doctor” debe ser ingresado por la restricción de valor no nulo.

¿Qué ocurre si la restricción de existencia la tuviéramos del lado del médico? Es decir, queremos que en el momento de dar de alta un médico se le debe asignar al menos un paciente.

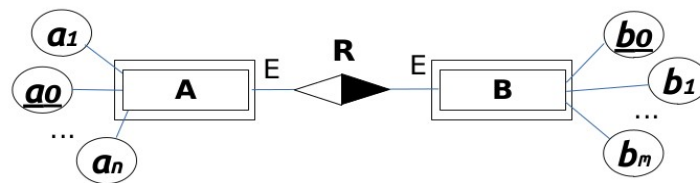


MEDICO			PACIENTE			
Ncolegiado	Nombre	Descripcion	Codigo	NSS	Nombre	Doctor
{Clave Primaria}	{Valor No Nulo}		{Clave Primaria}	{VNN, Único}	{Valor No Nulo}	{Valor No Nulo}
4600252125	ANA GIL	~~~~~	P0002	~~~~~	Jose Alberto Ruiz	NULL
4680025003	PEP SANZ	~~~~~	P1112	~~~~~	Ana M.ª Soler	NULL
4620360001	EVA BAS	~~~~~	P8886	~~~~~	Carlos Andrés Pau	4680025003
4611223344	JON ROS	~~~~~	P7773	~~~~~	Jose M.ª Gomez	4680025003
			P3333	~~~~~	M.ª Carmen Bens	4680025003
			P5550	~~~~~	Julio José Oliver	4620360001

Bueno, en este caso al tratar por ejemplo de dar de alta a un nuevo médico como por ejemplo a “Jon Ros” se debe representa esta restricción de existencia comprobando en la relación PACIENTE que en efecto se le asigne un paciente al menos, que es lo que se desea. **Se requiere pues dos acciones**, insertar al nuevo médico y asignarle un paciente. Para esto cabe definir la expresión presentada anteriormente con la que se comprobará que el nuevo médico tiene asignado un paciente o dicho de otro modo, aparece en la relación PACIENTE vinculado a al menos un paciente.

```
CREATE ASSERTION Rest_existencia CHECK
  NOT EXISTS ( SELECT * FROM MEDICO Mx
               WHERE NOT EXISTS (SELECT * FROM PACIENTES Px
                                WHERE Mx.Ncolegiado=Px.Doctor))
```


2.2.1.6 RELACIÓN BINARIA 1:M CON DOS RESTRICCIONES DE EXISTENCIA



Las relaciones equivalentes a esta estructura son:

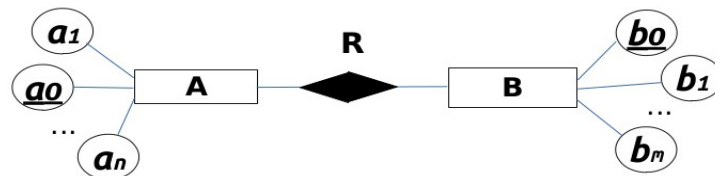
- A** (a0: dom_a0, a1: dom_a1,..., an: dom_an)
Clave Primaria: {a0}
- B** (b0: dom_b0, b1: dom_b1,..., bm: dom_bm, a0: dom_a0)
Clave Primaria: {b0}
Clave Ajena: {a0} hace referencia a A
Valor No Nulo: {a0}

Como puede apreciarse, este esquema es idéntico al de caso anterior, por tanto la restricción de existencia sobre la entidad A no está representada. Esta restricción habría que representarla mediante una expresión del SQL, del mismo modo que se ha visto en el caso anterior:

```
CREATE ASSERTION Rest_existencia CHECK
  NOT EXISTS (SELECT * FROM A Ax
    WHERE NOT EXISTS (SELECT * FROM B Bx
      WHERE Bx.a0=Ax.a0))
```

El diseño del caso de una relación binaria 1:M con una única restricción de existencia sobre la entidad con cardinalidad máxima n se realiza de forma equivalente al caso de una relación binaria 1:M sin restricciones de existencia, siendo por tanto también necesaria en este caso la utilización de una expresión del SQL para representar de nuevo la restricción de existencia.

2.2.1.7 RELACIÓN BINARIA M:M SIN RESTRICCIONES DE EXISTENCIA



Las relaciones equivalentes a esta estructura son:

- A** (a0: dom_a0, a1: dom_a1,..., an: dom_an)
Clave Primaria: {a0}
- B** (b0: dom_b0, b1: dom_b1,..., bm: dom_bm)
Clave Primaria: {b0}

R (a0: dom_a0, b0: dom_b0)

Clave Primaria: {a0, b0}

Clave Ajena: {a0} hace referencia a A

Clave Ajena: {b0} hace referencia a B

En este caso, debido a que las dos cardinalidades máximas son n, es necesario la representación de la relación binaria R mediante una relación independiente. La clave primaria de esta relación es el par {a0, b0} ya que es el único conjunto de atributos de R que satisface las condiciones exigidas a una clave primaria (Repásense estas condiciones en el libro de la asignatura de Bases de datos.)

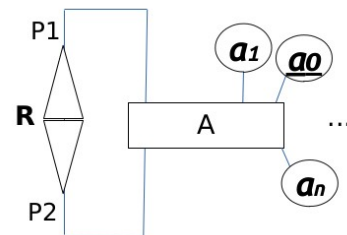
Los casos de relaciones binarias M:M en los que aparecen restricciones de existencia se diseñan de manera análoga, siendo necesario escribir expresiones en SQL que representen las restricciones de existencia.

2.2.2 RELACIONES REFLEXIVAS

Para entender las transformaciones de estas relaciones se recomienda compararlas con las binarias normales de igual cardinalidad ya que la estrategia seguida es la misma.

2.2.2.1 RELACIÓN BINARIA REFLEXIVA 1:1 SIN RESTRICCIÓN DE EXISTENCIA

El diagrama muestra una relación de este tipo. En la figura, p1 y p2 son dos etiquetas que indican el papel que cada ocurrencia de A juega en la relación.



La relación equivalente a esta estructura es:

A (a0: dom_a0, a1: dom_a1,..., an: dom_an, a0_p2: dom_a0)

Clave Primaria: {a0}

Único: {a0_p2}

Clave Ajena: {a0_p2} hace referencia a A

Dado que estas relaciones son difíciles de entender, a continuación se aclara su significado con un ejemplo. Sea una instancia del esquema anterior:

Cliente (cod: dom_cod, nom: dom_nombre, fecha_nac: dom_fecha, avala: dom_cod)

Clave Primaria: {cod}

Único: {avala}

Clave Ajena: {avala} hace referencia a Cliente

Cliente			
cod	nom	fecha_nac	Avala {Único}
FGR1	Fernando Gil Romero	20/10/1971	null
ASR1	Ana M.ª Soler Carrillo	08/05/1980	FGR1
JOB1	Julia Ortega Bas	03/06/1998	JSG1
JSG1	Jose Alberto Soriano García	07/01/2006	null

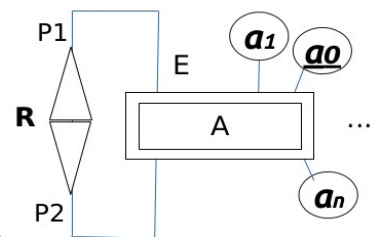
Vemos cuatro registros de tipo Cliente identificados por FGR1, ASR1, JOB1 y JSG1. Hay dos instancias de la relación R, representadas en las dos tuplas centrales, una que relaciona al Cliente ASR1 (jugando el papel p1, digamos “es avalado”) con el Cliente FGR1 (jugando el papel p2, digamos “avala”); y otra que relaciona al cliente JOB1 (jugando el papel p1, digamos “es avalado”) con el cliente JSG1 (jugando el papel p2, el que avala).

Así pues, en este caso, la relación R queda representada mediante la clave ajena *avala* en la relación que representa a la entidad.

Esta clave ajena también se define con restricción de unicidad para representar que la cardinalidad máxima de la entidad Cliente jugando el papel p2 es uno.

2.2.2.2 RELACIÓN BINARIA REFLEXIVA 1:1 CON RESTRICCIÓN DE EXISTENCIA

El diagrama muestra una relación de este tipo.



La relación equivalente a esta estructura es:

A (a0: dom_a0, a1: dom_a1,..., an: dom_an, a0_p2: dom_a0)

Clave Primaria: {a0}

Único: {a0_p2}

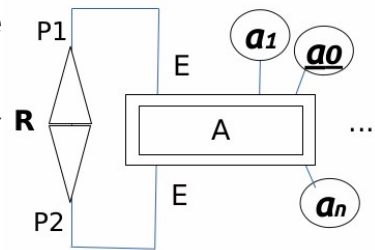
Clave Ajena: {a0_p2} hace referencia a A

Valor No Nulo: {a0_p2}

En este caso, la restricción de existencia se representa con la definición de la restricción de valor no nulo sobre la clave ajena a0_p2 .

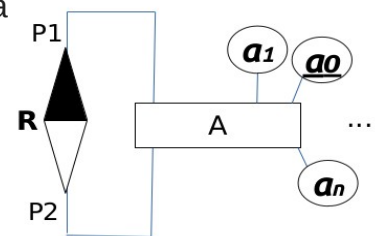
Cliente			
cod	nom	fecha_nac	Avala (Único, VNN)
FGR1	Fernando Gil Romero	20/10/1971	ASR1
ASR1	Ana M.ª Soler Carrillo	08/05/1980	FGR1
JOB1	Julia Ortega Bas	03/06/1998	JSG1
JSG1	Jose Alberto Soriano García	07/01/2006	JOB1

Curiosamente, este diagrama entidad-relación puede representar exactamente la misma realidad que el siguiente por lo que su transformación dará lugar a la misma relación.



2.2.2.3 RELACIÓN REFLEXIVA 1:M SIN RESTRICCIÓN DE EXISTENCIA

La figura muestra una relación de este tipo y la entidad asociada por la relación.



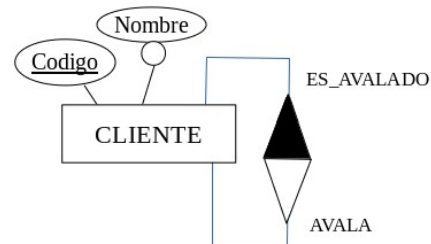
La relación equivalente a esta estructura es:

A (**a0**: dom_a0, a1: dom_a1,..., an: dom_an, a0_p2: dom_a0)
 Clave Primaria: {a0}
 Clave Ajena: {a0_p2} hace referencia a A

Veamos el siguiente ejemplo.

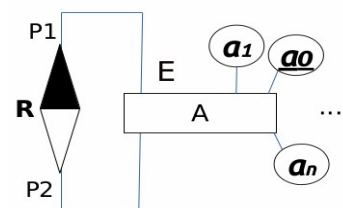
CLIENTE (Codigo: dom_cod, Nombre: dom_nom, Cod_aval: dom_cod)
 Clave Primaria: {Codigo}
 Clave Ajena: {Cod_aval} hace referencia a CLIENTE

CLIENTE		
Codigo	Nombre	Cod_Avala
{Clave Primaria}		
CLI01	Ana Gil	BTR12
CLI08	Jose Bas	BTC6
CLI25	Eva Sanz	NULL
CLI77	Jon Baez	NULL
		(Clave Ajena)



2.2.2.4 RELACIÓN REFLEXIVA 1:M CON UNA RESTRICCIÓN DE EXISTENCIA

Si por ejemplo tuviésemos una restricción de existencia es el sentido en el que “Al dar de alta un CLIENTE este avalará a otro cliente obligatoriamente” entonces tendríamos el siguiente caso:



La relación equivalente a esta estructura es:

A (a0: dom_a0, a1: dom_a1,..., an: dom_an, a0_p2: dom_a0)

Clave Primaria: {a0}

Clave Ajena: {a0_p2} hace referencia a A

Valor No Nulo {a0_p2}

Veámoslo sobre el ejemplo:

CLIENTE (Codigo: dom_cod, Nombre: dom_nom, Cod_aval: dom_cod)

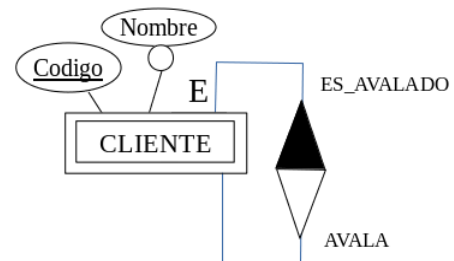
Clave Primaria: {Codigo}

Clave Ajena: {Cod_aval} hace referencia a CLIENTE

Valor No Nulo {Cod_aval}

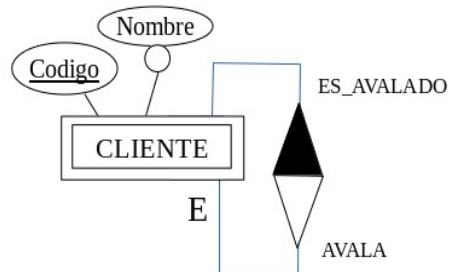
CLIENTE		
Codigo	Nombre	Cod_Avala
{Clave Primaria}		{VNN}
CLI01	Ana Gil	CLI08
CLI08	Jose Bas	CLI01
CLI25	Eva Sanz	CLI01
CLI77	Jon Baez	CLI01

{Clave Ajena}



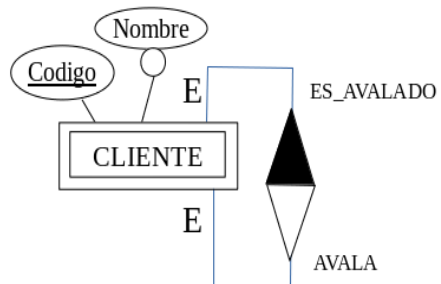
Pero, ¿cómo quedaría representada la siguiente idea?

“Un cliente debe ser avalado por al menos otro cliente.”



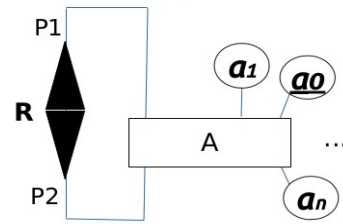
La solución anterior es también válida para este caso.

La siguiente idea con una doble restricción de existencia también quedaría representada de la forma anterior.



2.2.2.5 RELACIÓN REFLEXIVA M:M SIN RESTRICCIONES DE EXISTENCIA

El diagrama presenta este tipo de relación junto con la entidad asociada.



La relación equivalente a esta estructura es:

A (**a0**: dom_a0, a1: dom_a1,..., an: dom_an)

Clave Primaria: {a0}

R (**a0_p1**: dom_a0, **a0_p2**: dom_a0)

Clave Primaria: {a0_p1, a0_p2}

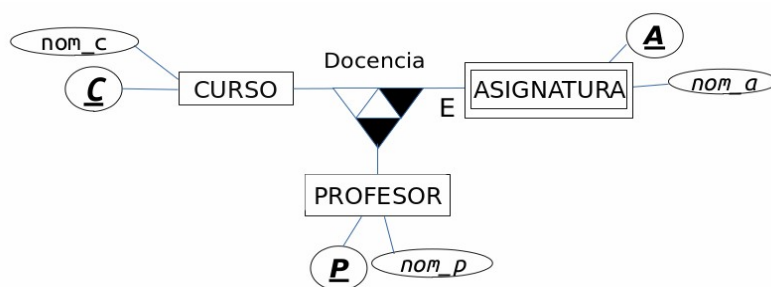
Clave Ajena: {a0_p1} hace referencia a A

Clave Ajena: {a0_p2} hace referencia a A

En este caso, debido a que la relación reflexiva R tiene cardinalidad máxima M:M, es necesario utilizar una relación independiente para representarla. Nótese cómo en este caso hay en R dos claves ajenas que se refieren a la misma relación, A. Si hubiera definida alguna restricción de existencia, también sería necesaria la inclusión de una expresión en SQL. La presencia de restricciones de existencia en las relaciones reflexivas es, sin embargo, poco frecuente.

2.2.3 RELACIÓN TERNARIAS

Para presentar este apartado empezaremos con un ejemplo:



Las relaciones que se derivan de este diseño son las siguientes:

CURSO (C: dom_C, nom_c: dom_nomc)

Clave Primaria: {C}

ASIGNATURA (A: dom_A, nom_a: dom_noma)

Clave Primaria: {A}

PROFESOR (P: dom_P, nom_p: dom_nomp)

Clave Primaria: {P}

DOCENCIA(C: dom_C, A: dom_A, P: dom_P)

Clave Primaria: {A, P}

Valor No Nulo: {C}

Clave Ajena: {C} hace referencia a curso

Clave Ajena: {A} hace referencia a asignatura

Clave Ajena: {P} hace referencia a profesor

Partimos de los siguientes datos:

CURSO	
C	nom_c
1DAM	Primero de DAM
2DAM	Segundo de DAM
1ASIR	Primero de ASIR
2ASIR	Segundo de ASIR

PROFESOR	
P	nom_p
AMG	Ana Montés Gil
RJG	Ramón Sanz García
RBT	Rosa Bas Torres
CGC	Carmen Gómez Coll

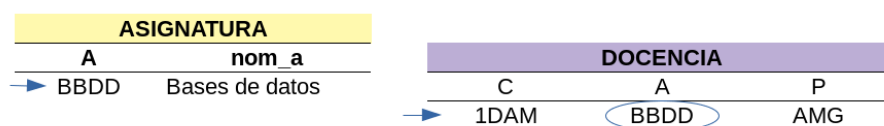
Las relaciones asignatura y docencia están todavía sin datos, pero existen asignaturas como BBDD: Bases de datos o PROG: Programación, ambas de primer curso de DAM.

En una relación ternaria la cardinalidad se define tomando relaciones de dos a una:

- Un par (1DAM, BBDD) puede estar relacionado con muchos profesores. Por ejemplo Ana y Rosa se encargan de ella.

- Un par (1DAM, AMG) puede estar relacionado con muchas asignaturas. Por ejemplo Ana en primero de DAM imparte varias asignaturas.
- Un par (AMG, BBDD) puede estar relacionado con UN SOLO curso. Ana solo dará BBDD en primero de DAM.

¿Qué ocurre en estos casos cuando, como en el ejemplo, tenemos una restricción de existencia? Para contestar a esto tratemos de ingresar en el sistema la asignatura de bases de datos (BBDD). Al tratar de hacerlo, la restricción de existencia nos dice que obligatoriamente debemos relacionar esta nueva asignatura con una ocurrencia dentro de la relación docencia. Con esto, nos damos cuenta que cuando una relación es de grado mayor que dos y presenta restricciones de este tipo, siempre serán necesarias definir restricciones SQL añadidas, definir una transacción ya que ingresar una ASIGNATURA implica ingresar en DOCENCIA otro registro.



Por otro lado, debemos saber que un elemento de una relación ternaria SIEMPRE tendrá un valor en cada entidad participante.

Nunca podríamos tener: 1DAM – null – AMG ¿Qué sentido tendría esto?

Nunca podríamos tener: null – BBDD – AMG o 1DAM – BBDD – null

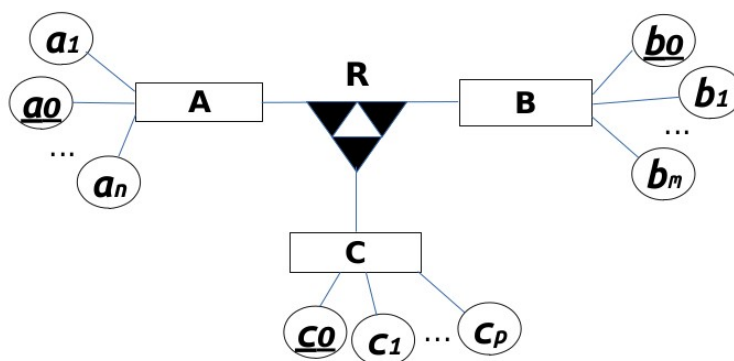
Comprueba como al definir **DOCENCIA** con Clave Primaria: {A, P} y Valor No Nulo: {C} evitamos cosas como las siguientes:

C (VNN)	C (CP)	C (CP)	
1DAM	BBDD	AMG	
1DAM	BBDD	RBT	! IMPOSIBLE ;
2DAM	BBDD	AMG	

Tras el razonamiento anterior, seguimos ahora mostrando cuatro casos de relaciones ternarias. En ninguno de ellos se considera la presencia de restricciones de existencia ya que cuando una relación es de grado mayor que dos siempre se representa independientemente de las entidades y son necesarias restricciones añadidas. Por otra parte, en todos los casos, con o sin restricción de existencia, la transformación de las entidades siguen el mismo patrón por lo que sólo se incluyen en el primer caso.

2.2.3.1 RELACIÓN TERNARIAS M:M:M

Este tipo de relación se muestra en el siguiente diagrama:

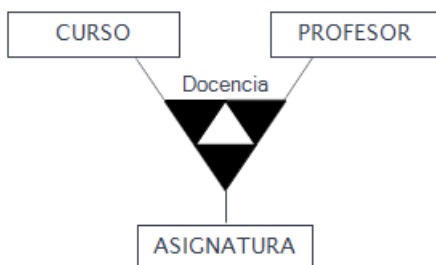


Las relaciones equivalentes a esta estructura son:

- A** (a0: dom_a0, a1: dom_a1,..., an: dom_an)
Clave Primaria: {a0}
- B** (b0: dom_b0, b1: dom_b1,..., bm: dom_bm)
Clave Primaria: {b0}
- C** (c0: dom_c0, c1: dom_c1,..., cp: dom_cp)
Clave Primaria: {c0}
- R** (a0: dom_a0, b0: dom_b0, c0: dom_c0)
Clave Primaria: {a0, b0, c0}
Clave Ajena: {a0} hace referencia a A
Clave Ajena: {b0} hace referencia a B
Clave Ajena: {c0} hace referencia a C

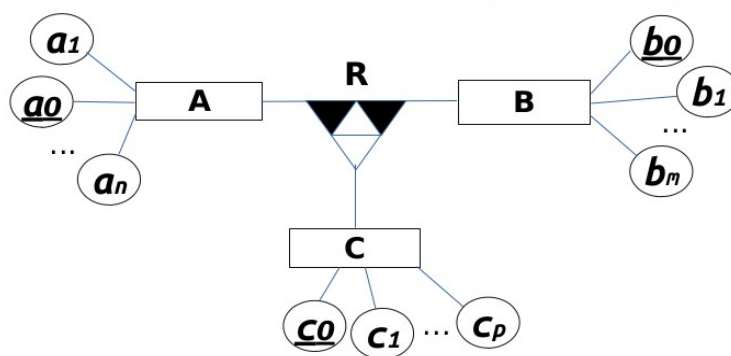
Para este tipo de cardinalidad máxima, la clave primaria de la relación R debe ser el atributo compuesto por las tres claves primarias de las tres relaciones que representan a las tres entidades.

EJEMPLO: DOCENCIA



M:M:M		
CP	CP	CP
PROFE	CURSO	ASIG
JoseMA	1º	MAT
XavierLV	1º	MAT
HugoGV	1º	FIS
XavierLV	1º	CAL
IsabelGG	1º	ING
JoseMA	1º	INF
CarolOR	1º	INF
SoniaOL	1º	FOL
HugoGV	2º	FIS
JoseMA	2º	FIS
JoseMA	2º	CAL
AnaJS	2º	ING
IsabelGG	2º	ING
AnaJS	2º	INF
CarolOR	2º	EST
XavierLV	2º	EST

2.2.3.2 RELACIÓN TERNARIAS 1:M:M



En este caso la relación R es la siguiente:

R (a0: dom_a0, b0: dom_b0, c0: dom_c0)

Clave Primaria: {a0, b0}

Valor No Nulo: {c0}

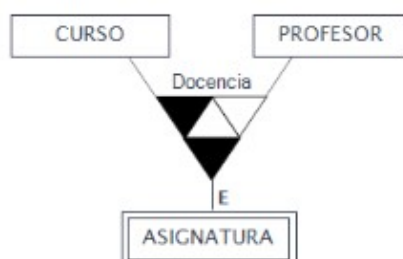
Clave Ajena: {a0} hace referencia a A

Clave Ajena: {b0} hace referencia a B

Clave Ajena: {c0} hace referencia a C

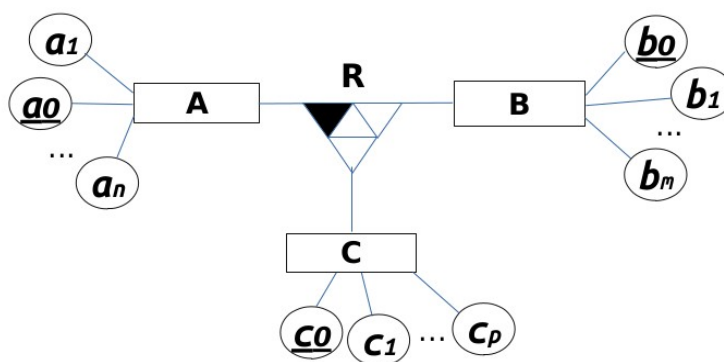
Hay que llamar la atención sobre la definición de una restricción de valor no nulo sobre el atributo c0, que es la clave primaria de la relación C. La razón para esto, es la necesidad de evitar que puedan aparecer tuplas de la relación R que tengan valor nulo para el atributo c0, ya que estas tuplas no representan ocurrencias reales de la relación R. Así, como regla general, se deberá definir una restricción de valor no nulo sobre todo atributo correspondiente a una clave ajena que no esté presente en la clave primaria de la relación que representa a la relación ternaria del esquema entidad-relación.

EJEMPLO: DOCENCIA



	1:M:M	
VNN	CP	CP
PROFE	CURSO	ASIG
JoseMA	1º	MAT
XavierLV	1º	MAT
HugoGV	1º	FIS
XavierLV	1º	CAL
IsabelGG	1º	ING
JoseMA	1º	INF
CarolOR	1º	INF
SoniaOL	1º	FOL
HugoGV	2º	FIS
JoseMA	2º	FIS
JoseMA	2º	CAL
AnaJS	2º	ING
IsabelGG	2º	ING
AnaJS	2º	INF
CarolOR	2º	EST
XavierLV	2º	EST

2.2.3.3 RELACIÓN TERNARIAS 1:1:M



En este caso la relación R es la siguiente:

R (a0: dom_a0, b0: dom_b0, c0: dom_c0)

Clave Primaria: {a0, b0}

Único: {a0, c0}

Valor No Nulo: {c0}

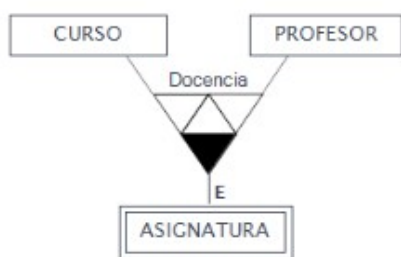
Clave Ajena: {a0} hace referencia a A

Clave Ajena: {b0} hace referencia a B

Clave Ajena: {c0} hace referencia a C

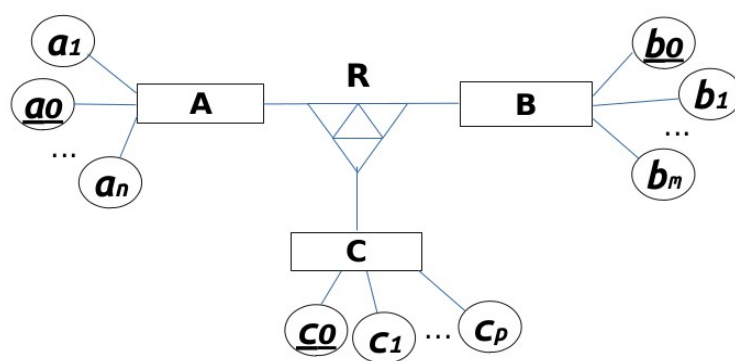
Debido a la presencia de dos cardinalidades máximas 1, existen dos claves candidatas para la relación R; cualquiera de las dos puede ser clave primaria, teniendo que definirse para la otra una restricción de unicidad, además de la correspondiente restricción de valor no nulo sobre la clave ajena que no forma parte de la clave primaria.

EJEMPLO: DOCENCIA



	1:1:M	
VNN	CP	CP
PROFE (UNICO)	CURSO	ASIG (UNICO)
JoseMA	1º	MAT
XavierLV	1º	MAT
HugoGV	1º	FIS
XavierLV	1º	CAL
IsabelGG	1º	ING
JoseMA	1º	INF
CarolOR	1º	INF
SoniaOL	1º	FOL
HugoGV	2º	FIS
JoseMA	2º	FIS
JoseMA	2º	CAL
AnaJS	2º	ING
IsabelGG	2º	ING
AnaJS	2º	INF
CarolOR	2º	EST
XavierLV	2º	EST

2.2.3.4 RELACIÓN TERNARIAS 1:1:1



En este caso la relación R es la siguiente:

R (a0: dom_a0, b0: dom_b0, c0: dom_c0)

Clave Primaria: {a0, b0}

Único: {a0, c0}

Único: {b0, c0}

Valor No Nulo: {c0}

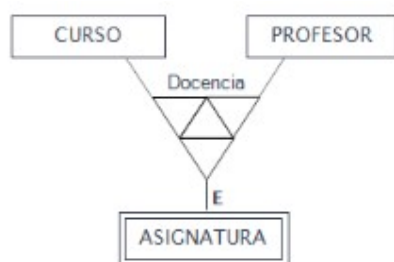
Clave Ajena: {a0} hace referencia a A

Clave Ajena: {b0} hace referencia a B

Clave Ajena: {c0} hace referencia a C

En este caso, existen tres cardinalidades máximas 1, por lo que hay tres claves candidatas para la relación R; cualquiera de las tres puede ser clave primaria teniendo que definirse para las otras dos una restricción de unicidad.

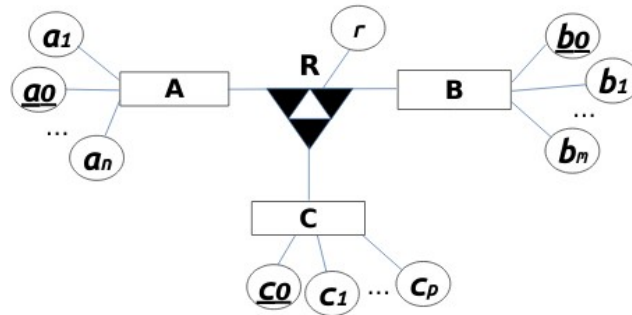
EJEMPLO: DOCENCIA



1:1:1		
VNN	CP	CP
PROFE (UNICO)	CURSO (UNICO)	ASIG (UNICO)
JoseMA	1º	MAT
XavierLV	1º	MAT
HugoGV	1º	FIS
XavierLV	1º	CAL
IsabelGG	1º	ING
JoseMA	1º	INF
CarolOR	1º	INF
SoniaOL	1º	FOL
HugoGV	2º	FIS
JoseMA	2º	FIS
JoseMA	2º	CAL
AnaJS	2º	ING
IsabelGG	2º	ING
AnaJS	2º	INF
CarolOR	2º	EST
XavierLV	2º	EST

2.3 ATRIBUTOS EN LAS RELACIONES

Cuando una relación entre entidades, por ejemplo R, tiene atributos propios, éstos deben incluirse siempre en la relación donde se represente R. Por ejemplo, sea el siguiente diagrama entidad-relación:

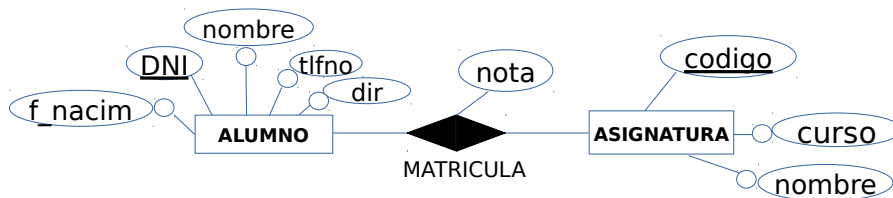


Las relaciones equivalentes a esta estructura son:

- A** (a0: dom_a0, a1: dom_a1,..., an: dom_an)
Clave Primaria: {a0}
- B** (b0: dom_b0, b1: dom_b1,..., bm: dom_bm)
Clave Primaria: {b0}
- C** (c0: dom_c0, c1: dom_c1,..., cp: dom_cp)
Clave Primaria: {c0}
- R** (a0: dom_a0, b0: dom_b0, c0: dom_c0, **r: dom_r**)
Clave Primaria: {a0, b0, c0}
Clave Ajena: {a0} hace referencia a A
Clave Ajena: {b0} hace referencia a B
Clave Ajena: {c0} hace referencia a C

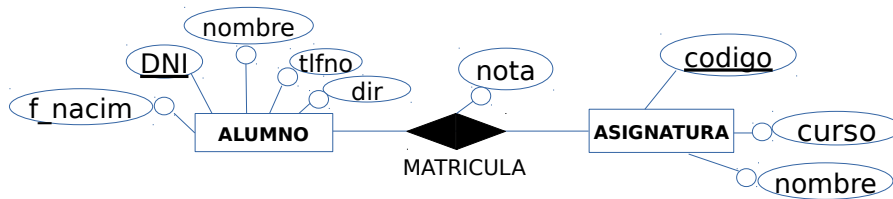
La presencia de atributos en las relaciones de tipo muchos a muchos son muy comunes y la resolución de su presencia pasa por lo que se ha expuesto.

EJEMPLO:



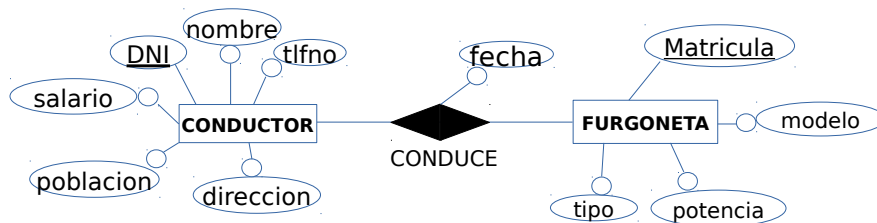
MATRICULA		
Alumno	Asignatura	Nota
{Clave Primaria}		
Raquel Valls	Programación I	null
José R. Coll	Programación I	8
Ferran Orts	Sistemas en Red	5
Gabriel Castro	Bases de datos	null

En el siguiente ejemplo el atributo es obligatorio.



MATRICULA		
Alumno	Asignatura	Nota
{Clave Primaria}		{Valor No Nulo}
Raquel Valls	Programación I	0
José R. Coll	Programación I	8
Ferran Orts	Sistemas en Red	5
Gabriel Castro	Bases de datos	0

Pero mucha atención a casos como el siguiente:



En este diagrama Entidad-Relación se pretende hacer notar que se desea registrar las veces que un conductor ha llevado diferentes furgonetas en diferentes fechas. Pero esto no es posible del modo que estamos representándolo.

CONDUCE		
Conductor	Furgoneta	Fecha
{Clave Primaria}		{Valor No Nulo}
Fernando Paz	9603-HYS	25/11/20
Eva Navarro	7610-JBB	19/11/20
Pablo Espi	5058-TRD	21/11/20
Pablo Espi	2020-HJH	17/11/20
Eva Navarro	7610-JBB	21/11/20

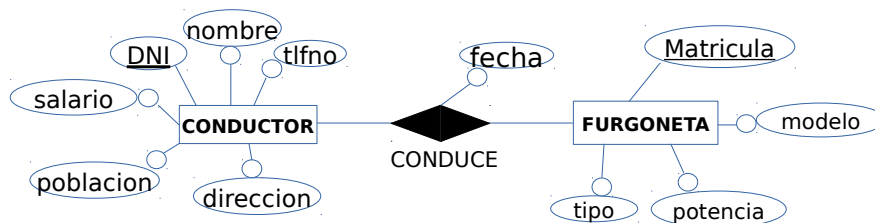
Como se observa en el ejemplo, solo podemos registrar a Eva conduciendo una furgoneta un día. Si quisiéramos representar lo deseado deberíamos poner la fecha formando parte de la clave primaria de la relación CONDUCE.

CONDUCE		
Conductor	Furgoneta	Fecha
{Clave Primária}		
Fernando Paz	9603-HYS	25/11/20
Eva Navarro	7610-JBB	19/11/20
Pablo Espi	5058-TRD	21/11/20
Pablo Espi	2020-HJH	17/11/20
Eva Navarro	7610-JBB	21/11/20

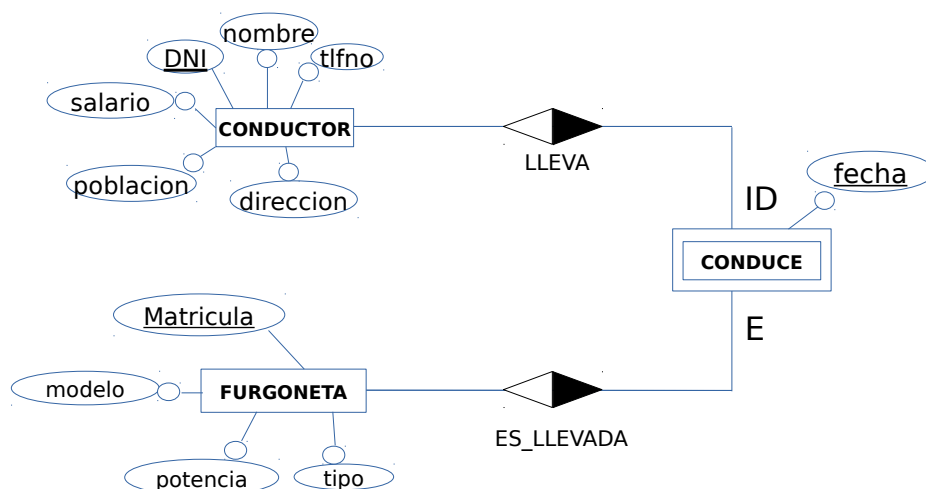
Bien, pero ahora no puedo dejar la fecha con valor nulo... ¿Tendría sentido?

Estrictamente y tal como se indicó en el tema anterior, si en una relación R entre las entidades E1,...,En puede existir más de una ocurrencia de R que relaciona las mismas ocurrencias de E1, ..., En, entonces la relación **está mal definida**. Esto es lo que ocurre en el ejemplo anterior. Estamos queriendo relacionar a Eva Navarro dos veces con la furgoneta de matricula 7610-JBB. En distinta fecha claro está.

Estrictamente la siguiente relación esta mal definida:



La siguiente es una aproximación más correcta.

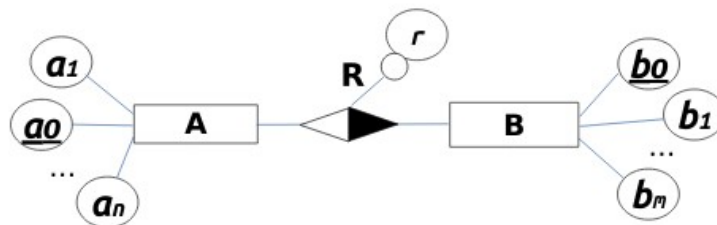


Así quedaría la relación CONDUCE.

CONDUCE		
Conductor	Fecha	Furgoneta
{Clave Primária}		{Valor No Nulo}
Fernando Paz	25/11/20	9603-HYS
Eva Navarro	19/11/20	7610-JBB
Pablo Espi	21/11/20	5058-TRD
Pablo Espi	17/11/20	2020-HJH
Eva Navarro	21/11/20	7610-JBB

Hemos visto los casos en relaciones de tipo muchos a muchos, pero la presencia de atributos en las relaciones de tipo uno a muchos puede suponer la aparición de restricciones de integridad que hagan que un esquema relacional que sería adecuado sin los atributos deje de serlo por su presencia. Esta situación se ilustra a continuación. Se considerarán tres casos distintos:

Caso a)



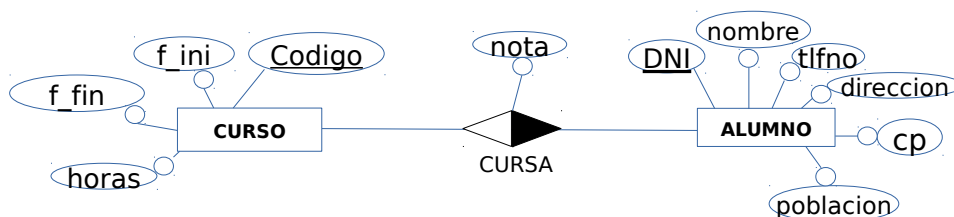
Este esquema quedaría así:

A (a0: dom_a0, a1: dom_a1,..., an: dom_an)
Clave Primaria: {a0}

B (b0: dom_b0, b1: dom_b1,..., bm: dom_bm, a0: dom_a0, r: dom_r)
Clave Primaria: {b0}
Clave Ajena: {a0} hace referencia a A

Veamos, sin embargo, como este caso es difícilmente aplicable en la realidad y en caso de darse deberemos tratarlo del siguiente modo.

Veamos un ejemplo.



Un alumno de este sistema cursa uno de los cursos de formación y solo uno. De la relación que se establece vemos que la nota es obligatoria.

ALUMNO				
DNI	nombre	Curso	nota
53503427E	Joana Ortiz		HS88BX041	6
50378992E	Ceferino Castro		FP85CM045	9
10859920X	Aranzazu Gómez			
39283654E	Tania Marín			

Tania y Aranzazu no tienen curso asignado.

¿VNN?

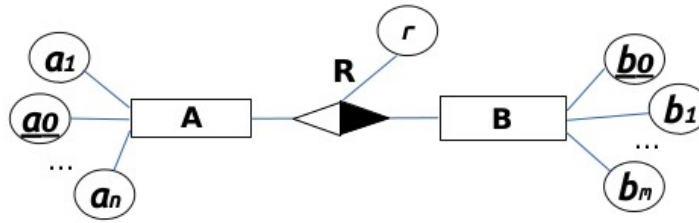
En este esquema no se puede especificar que el atributo **nota** tiene restricción de valor no nulo **ya que no todas las ocurrencias de ALUMNO tienen que tener valor en el atributo nota sino sólo aquellas que se relacionan con una ocurrencia de CURSO**. Así pues, para que este esquema relacional sea correcto se debe añadir la siguiente restricción de integridad:

```
CREATE ASSERTION atributo_no_nulo_en_relación CHECK
  NOT EXISTS ( SELECT * FROM ALUMNO AI
    WHERE (AI.Curso IS NOT NULL AND AI.nota IS NULL )
    OR
    (AI.Curso IS NULL AND AI.nota IS NOT NULL ))
```

Al añadir una restricción de integridad, este esquema relacional deja de ser el más adecuado ya que hay otro que, aunque tiene una relación más, no necesita restricciones añadidas. Este esquema es el siguiente:

- A** (a0: dom_a0, a1: dom_a1,..., an: dom_an)
Clave Primaria: {a0}
- B** (b0: dom_b0, b1: dom_b1,..., bm: dom_bm)
Clave Primaria: {b0}
- R** (a0: dom_a0, b0: dom_b0, r: dom_r)
Clave Primaria: {b0}
Valor No Nulo: {a0}
Valor No Nulo: {r}
Clave Ajena: {a0} hace referencia a A
Clave Ajena: {b0} hace referencia a B

Caso B)



Las relaciones que se obtendrían siguiendo la transformación general serían las mismas que en el caso anterior, pero habría que incluir la restricción de integridad siguiente:

```
CREATE ASSERTION atributo_en_relación CHECK
    NOT EXISTS ( SELECT * FROM B Bx
        WHERE (Bx.a0 IS NULL AND Bx.r IS NOT NULL))
```

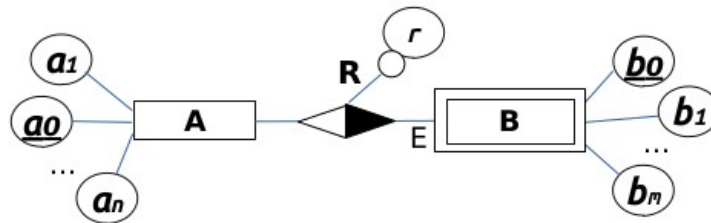
por lo que también en este caso es más adecuado el siguiente esquema:

A (a0: dom_a0, a1: dom_a1,..., an: dom_an)
Clave Primaria: {a0}

B (b0: dom_b0, b1: dom_b1,..., bm: dom_bm)
Clave Primaria: {b0}

R (a0: dom_a0, b0: dom_b0, r: dom_r)
Clave Primaria: {b0}
Valor No Nulo: {a0}
Clave Ajena: {a0} hace referencia a A
Clave Ajena: {b0} hace referencia a B

Caso C)

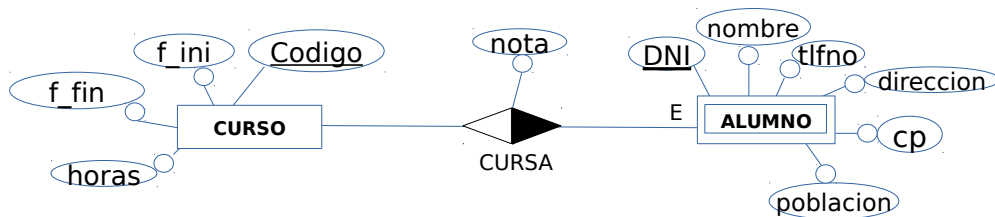


A (a0: dom_a0, a1: dom_a1,..., an: dom_an)
Clave Primaria: {a0}

B (b0: dom_b0, b1: dom_b1,..., bm: dom_bm, a0: dom_a0, r: dom_r)
Clave Primaria: {b0}
Valor No Nulo: {a0}
Valor No Nulo: {r}
Clave Ajena: {a0} hace referencia a A

En esta situación en la que la entidad B tiene restricción de existencia respecto a R, dado que, sobre el atributo a0 de la relación B resultante se define una

restricción de valor no nulo, lo más adecuado es representarlo como se ha hecho en los casos en los que no había atributos en la relación, es decir, la relación R junto a la entidad B, tanto si el atributo r tiene restricción de valor no nulo como si no.



CURSO (Codigo, f_ini, f_fin, horas)

Clave Primaria: {Codigo}

ALUMNO (DNI, nombre, tfno, direccion, cp, poblacion, curso, nota)

Clave Primaria: {DNI}

Valor No Nulo: {curso}

Valor No Nulo: {nota}

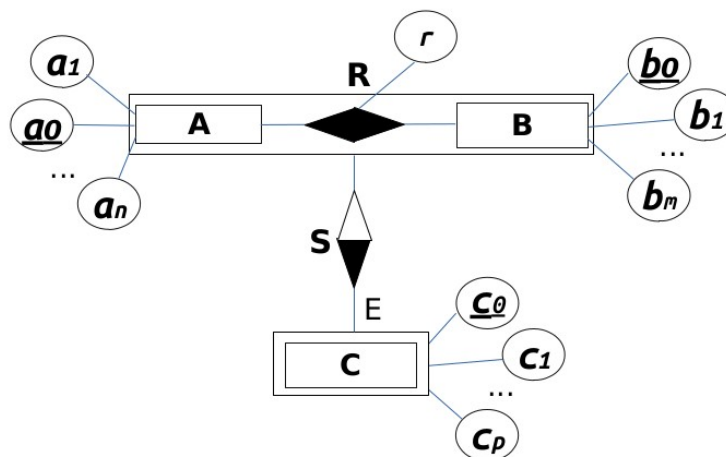
Clave Ajena: {curso} hace referencia a CURSO

ALUMNO				
DNI	nombre	curso	nota
53503427E	Joana Ortiz		HS88IN050	8
50378992E	Ceferino Castro		GA47IN017	5
10859920X	Aranzazu Gómez		AI32CF011	0
39283654E	Tania Marín		OR99IN047	0

curso y nota son campos obligatorios.

Podemos poner un valor predeterminado hasta conocer la nota final.

2.4 TRANSFORMACIÓN DE LA AGREGACIÓN



Generalizar el diseño de los objetos agregados que pueden aparecer en un diagrama entidad-relación es complicado ya que éstos pueden estar definidos sobre relaciones de cualquier grado y, por otro lado, tomar parte en cualquier otra relación del diagrama. Por ello se verá primero cómo se realiza este diseño para un caso concreto y luego se dará una guía general para cualquier tipo de agregación.

Supongamos que el diagrama de la figura anterior es el diagrama a diseñar. Como este diagrama contiene una agregación, se deberá iniciar el diseño transformando la parte del diagrama en la que está representada la agregación. Las relaciones que se obtienen son:

- A** (**a0**: dom_a0, a1: dom_a1,..., an: dom_an)
Clave Primaria: {a0}
- B** (**b0**: dom_b0, b1: dom_b1,..., bm: dom_bm)
Clave Primaria: {b0}
- R** (**a0**: dom_a0, **b0**: dom_b0, r: dom_r)
Clave Primaria: {a0, b0}
Clave Ajena: {a0} hace referencia a A
Clave Ajena: {b0} hace referencia a B

Claramente, la relación que está representando al objeto agregado es R, ya que cada tupla de la relación R expresa que una ocurrencia de la entidad A y otra de B están asociadas. La clave de esta relación es pues la clave o identificador del objeto agregado en el esquema relacional. En nuestro ejemplo esta clave es el atributo compuesto {a0, b0}. Esta clave es el identificador del objeto agregado y deberá ser usada como tal cuando se diseñen las relaciones en las cuales toma parte el objeto agregado.

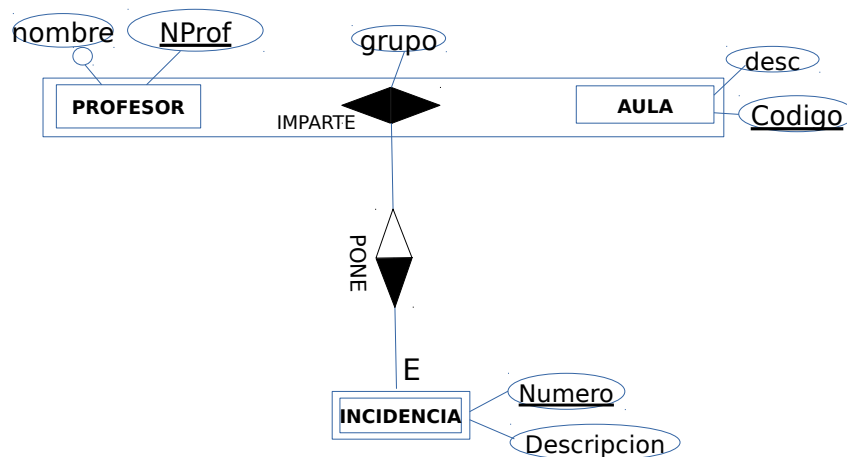
El segundo paso consiste en transformar el resto del diagrama, es decir, las relaciones en las cuales toman parte el objeto agregado y el resto de entidades y relaciones del diagrama. En el ejemplo anterior, consiste en transformar la relación **S** en la cual toman parte el objeto agregado y la entidad C, y diseñar también la propia entidad C.

Ya que la relación S es una relación binaria entre el objeto agregado y C con cardinalidad máxima 1:M y C sufre una restricción de existencia, se debe obtener la relación siguiente (ver apartado 2.2.1.5):

- C** (**c0**: dom_c0, c1: dom_c1,..., cp: dom_cp, **a0**: dom_a0, **b0**: dom_b0)
Clave Primaria: {c0}
Clave Ajena: {a0, b0} hace referencia a R
Valor No Nulo: {a0, b0}

Esta relación representa tanto la entidad C como la relación S. Esta última se representa mediante una clave ajena {a0, b0} que hace referencia a la relación en la cual está representado el objeto agregado, es decir R.

Vamos a desarrollar con un ejemplo la agregación anterior.



PROFESOR (NProf: dom_numprof, nombre: dom_nom)
Clave Primaria: {NProf}

AULA (Codigo: dom_codAula, desc: dom_desc)
Clave Primaria: {Codigo}

IMPARTE (NProf: dom_num, Codigo: dom_codA, grupo: dom_grupo)
Clave Primaria: {a0, b0}
Clave Ajena: {a0} hace referencia a A
Clave Ajena: {b0} hace referencia a B

INCIDENCIA (Numero: dom_n, Descripcion: dom_des, NProf: dom_num, Codigo: dom_codA)
Clave Primaria: {Numero}
Clave Ajena: {a0, b0} hace referencia a R
Valor No Nulo: {a0, b0}

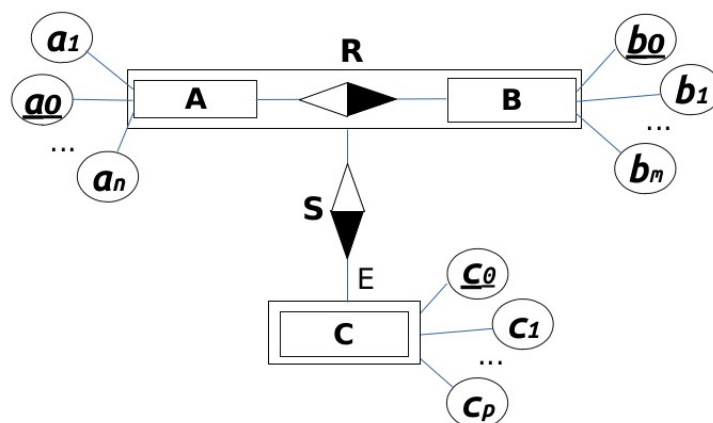
PROFESOR			
NProf	nombre		
{Clave Primaria}			
10	Raquel Valls		
21	José R. Coll		
15	Ferran Orts		
8	Gabriel Castro		

AULA			
Codigo	Tipo		
{Clave Primaria}			
A13	Aula pasillo B1º		
A14	Aula pasillo B2º		
A15	Aula pasillo B3º		
A7	Aula pasillo A		

IMPARTE				
NProf	Codigo	grupo		
{Clave Primaria}				
10	A13	2SMR		
10	A15	1DAM		
21	A13	2SMR		
15	A13	1ASIR		
21	A14	1ASIR		
{Clave Ajena}	{Clave Ajena}			

INCIDENCIA			
Numero	Descripcion	NProf	Codigo
{Clave Primaria}			
1	PC 12 no arranca 14/11/19	10	A15
2	Luz 3ª fila fundida	10	A15
3	Persiana ventana fondo rota	15	A13
4	Teclado PC2 faltan teclas	10	A15
5	Instalar Pizarra digital	21	A14
6	PC 8 clonar imagen disco	10	A15
		{Clave Ajena}	

En otros casos la agregación de una relación puede hacernos reconsiderar el diseño propuesto para la relación sin agregar. Esto se ilustra en el siguiente ejemplo:



Las relaciones que se obtienen al transformar la parte superior del diagrama, teniendo en cuenta las normas explicadas en el apartado 2.2.1.4 son las siguientes:

- A** (a0: dom_a0, a1: dom_a1,..., an: dom_an)
 Clave Primaria: {a0}
- B** (b0: dom_b0, b1: dom_b1,..., bm: dom_bm, a0: dom_a0)
 Clave Primaria: {b0}
 Clave Ajena: {a0} hace referencia a A

Ahora es necesario identificar qué relación está representando al objeto agregado. En este caso no hay una relación independiente sino que R está representada en B. El segundo paso consiste como antes en transformar el resto del diagrama.

- C** (c0: dom_c0, c1: dom_c1,..., cp: dom_cp, b0: dom_b0)
 Clave Primaria: {c0}
 Clave Ajena: {b0} hace referencia a B
 Valor No Nulo: {b0}

Ahora bien, este esquema no es adecuado ya que podría darse el caso de que una ocurrencia de C se relacionara con una ocurrencia de B que no se relaciona con ninguna ocurrencia de A, por lo que para que el esquema fuese correcto habría que añadir la siguiente restricción de integridad:

```
CREATE ASSERTION Agregado CHECK
  NOT EXISTS ( SELECT Cx.c0 FROM C Cx WHERE
    EXISTS ( SELECT Bx.b0 FROM B Bx
      WHERE (Bx.b0 =Cx.b0 AND Bx.a0 IS NULL)))
```

De nuevo la aparición de la restricción hace que este esquema no sea el más deseable ya que se puede encontrar otro con una relación más pero sin restricciones añadidas. Este esquema es el que se muestra:

A (a0: dom_a0, a1: dom_a1,..., an: dom_an)
Clave Primaria: {a0}

B (b0: dom_b0, b1: dom_b1,..., bm: dom_bm)
Clave Primaria: {b0}

R (a0: dom_a0, b0: dom_b0)
Clave Primaria: {b0}
Valor No Nulo: {a0}
Clave Ajena: {a0} hace referencia a A
Clave Ajena: {b0} hace referencia a B

C (c0: dom_c0, c1: dom_c1,..., cp: dom_cp, b0: dom_b0)
Clave Primaria: {c0}
Clave Ajena: {b0} hace referencia a R
Valor No Nulo: {b0}

Resumiendo, la guía para realizar el diseño de los objetos agregados se puede concretar en los tres siguientes puntos:

- 1) Realizar la transformación del subesquema correspondiente al objeto agregado. En el supuesto de que existieran varios, se realizaría el diseño del más simple al más complejo. En este punto hay que tener en cuenta los problemas que pueden aparecer si la relación que se agrega se representa junto con alguna entidad como se ha ilustrado en el ejemplo.
- 2) Identificar la relación que representa al objeto agregado y la clave de la misma en las relaciones obtenidas en el punto anterior.
- 3) Continuar con la transformación del resto del diagrama. Las participaciones del objeto agregado en relaciones se resolverán teniendo en cuenta el punto anterior, es decir, dónde se encuentra representado el objeto agregado y cuál es su clave.

2.5 RESUMEN

El conjunto de transformaciones que se ha presentado permite traducir cualquier objeto de un diagrama entidad-relación a un conjunto de definiciones de relaciones, y en su caso, de expresiones en SQL (que expresarían las propiedades del diagrama E-R que no tienen una transformación exacta al modelo relacional). Los siguientes puntos constituyen una guía para la aplicación de estas reglas:

- 1) Es aconsejable, iniciar la transformación por las entidades fuertes y después transformar el resto de entidades. En estas relaciones se definirán también, cuando sea posible, las relaciones con al menos una cardinalidad máxima a 1. Una vez estén todas las entidades transformadas a relaciones en el esquema relacional, continuar por las demás relaciones del diagrama.
- 2) En presencia de objetos agregados iniciar la transformación por los objetos agregados (del más interior al más exterior). Una vez estén los objetos agregados transformados y sabiendo dónde se encuentran representados en el esquema relacional obtenido hasta ese momento, continuar la transformación del resto del diagrama. Para cualquier participación de un objeto agregado en una relación se deberá utilizar la representación de ese objeto que se identificó en el paso anterior.