

EJEMPLO DE INGENIERÍA INVERSA DE DATOS*

Miguel-Angel Sicilia

This work is produced by The Connexions Project and licensed under the Creative Commons Attribution License †

Abstract

Ejemplo práctico de realizar un análisis de base de datos para aplicar Ingeniería Inversa de Datos.

La Ingeniería Inversa de Bases de Datos es el conjunto de técnicas que permite la obtención de una representación conceptual de un esquema de base de datos a partir de su codificación.

Sus aplicaciones son múltiples: Re-documentar, reconstruir y/o actualizar documentación perdida o inexistente de bases de datos, servir como pivote en un proceso de migración de datos, y ayudar en la exploración y extracción de datos en bases poco documentadas.

Ahora se comienza a realizar el análisis por el cual obtendremos el modelo conceptual de una base de datos a partir de un modelo físico.

Esta aplicación está implementada en Delphi, con un Oracle Server 8i Lite, por lo tanto los ejemplos están basados en dichos productos. De todas formas, el análisis es el mismo a seguir independientemente del lenguaje o base de datos que utilicemos.

Lo primero que se debe de hacer es obtener toda la información posible de la estructura de la base de datos (no de los datos que contiene), es decir, nombre de las tablas, atributos de las tablas, etc. Dicha información se encuentra almacenada en el catálogo de la base de datos (el cual se consulta fácilmente utilizando SQL). La información obtenida a partir del catálogo se debe almacenar en algún lado (se suele crear una serie de clases que permitan almacenar toda la información y además a dichas clases se les agrega cierta funcionalidad que permita manejar fácilmente la información almacenada en ellas).

Para realizar la obtención de todas las tablas que componen la base de datos se debe efectuar una consulta SQL. En dicha consulta se obtiene los nombres de las tablas, atributos que componen las tablas con sus características más generales (tipos de datos y si admite valores nulos).

La consulta SQL que utilice es la siguiente:

```
SELECT at.table_name, attc.column_name, attc._data_type, attc.nullable
FROM all_tables at, all_tab_columns attc
WHERE at.table_name = attc.table_name
```

El resultado de dicha consulta será el siguiente: por cada fila habrán cuatro columnas. Las columnas significan lo siguiente: nombre de la tabla, nombre del atributo, tipo de dato del atributo y si el atributo puede ser nulo. Por lo tanto, cada tabla tendrá tantas filas en el resultado de la consulta como atributos posea.

Una vez realizada esta consulta se procede a guardarla en las estructuras de almacenamiento. Una vez hecho se puede analizar cuales atributos de las tablas corresponden a la clave primaria, cuales son claves foráneas y cuales son claves únicas (que en el modelo de normalización serían las claves candidatas).

*Version 1.3: Nov 25, 2008 8:59 am -0600

†<http://creativecommons.org/licenses/by/2.0/>

Para obtener aquellos atributos que componen la clave primaria de una tabla dada se realiza la siguiente consulta, que se debe realizar para cada tabla existente en la base de datos, cambiando en la siguiente consulta NombreTabla por el nombre de la tabla que se consulta (a partir de este momento, cada vez que se coloque NombreTabla se entenderá que es la tabla que nos encontramos analizando). Aquí va la consulta SQL realizada:

```
SELECT column_name
FROM all_constraints ac, all_cons_columns acc
WHERE ac.table_name = 'NombreTabla'
AND ac.constraint_type = 'P'
AND ac.constraint_name = acc.constraint_name
ORDER BY acc.position;
```

El resultado de esta consulta es una fila para cada atributo que forma parte de la clave primaria. Dichos atributos se desplegarán en orden ascendente según su posición, para así poder ingresarlos en el orden por el cual fueron definidos.

A continuación se obtendrán los atributos que forman las claves foráneas de una tabla, y las tablas a las cuales hace referencia dicha clave foránea perteneciente a una determinada tabla, que se llamará nuevamente NombreTabla.

```
SELECT ac.constraint_name, column_name, r_constraint_name
FROM all_constraints ac, all_cons_columns acc
WHERE ac.table_name = 'NombreTabla'
AND ac.constraint_type = 'R'
AND ac.constraint_name = acc.constraint_name
ORDER BY acc.position;
```

Como resultado se obtiene una fila por cada atributo que compone a una clave foránea. Cada constraint (clave foránea en este caso) tendrá tantas filas como atributos los compongan. Por cada columna tenemos la siguiente información en el siguiente orden: nombre del constraint, nombre del atributo y nombre del constraint al cual hace referencia.

A partir de los constraints a los cuales se hacen referencia, se puede obtener fácilmente a que tabla pertenecen por medio de la siguiente consulta:

```
SELECT table_name
FROM all_constraints
WHERE constraint_name = 'NombreConstraint'
```

Con la consulta anterior obtenemos a que tabla pertenece el constraint NombreConstraint y por lo tanto, si una clave foránea hace referencia al constraint NombreConstraint, entonces ahora sabemos a que tabla hace referencia dicha clave foránea.

Finalmente, para la carga de datos sólo falta averiguar cuales son los atributos que componen a las claves únicas. Para esto se realiza la siguiente consulta:

```
SELECT ac.constraint_name, column_name
FROM all_constraints ac, all_cons_columns acc
WHERE ac.table_name = 'NombreTabla'
AND ac.constraint_type = 'U'
AND ac.constraint_name = acc.constraint_name
ORDER BY acc.position;
```

La consulta anterior devuelve todas las claves únicas que existen en una tabla. Cada clave única tendrá tantas filas en el resultado de la consulta como atributos la compongan. El significado de las columnas es el siguiente: nombre del constraint (o sea, de la clave única en este caso) y nombre del atributo.

1 Análisis de las tablas

A continuación se presenta como determinar que representación conceptual tiene una tabla dada. Es decir, por ejemplo, una tabla puede ser considerada una entidad, una relación binaria, una relación ternaria, una

categorización, etc.

En general, el siguiente análisis debe ser realizado por todas las tablas. Como guía, se presenta un diagrama de flujo, que es el que se debe seguir a la hora de analizar una tabla. Es decir, a una tabla dada se le realizarán ciertas pruebas y en función de los resultados de dichas pruebas decidiremos que 'camino' del diagrama de flujo seguir.

A continuación se presenta el diagrama de árbol que sirve de ayuda a la hora de realizar el análisis.

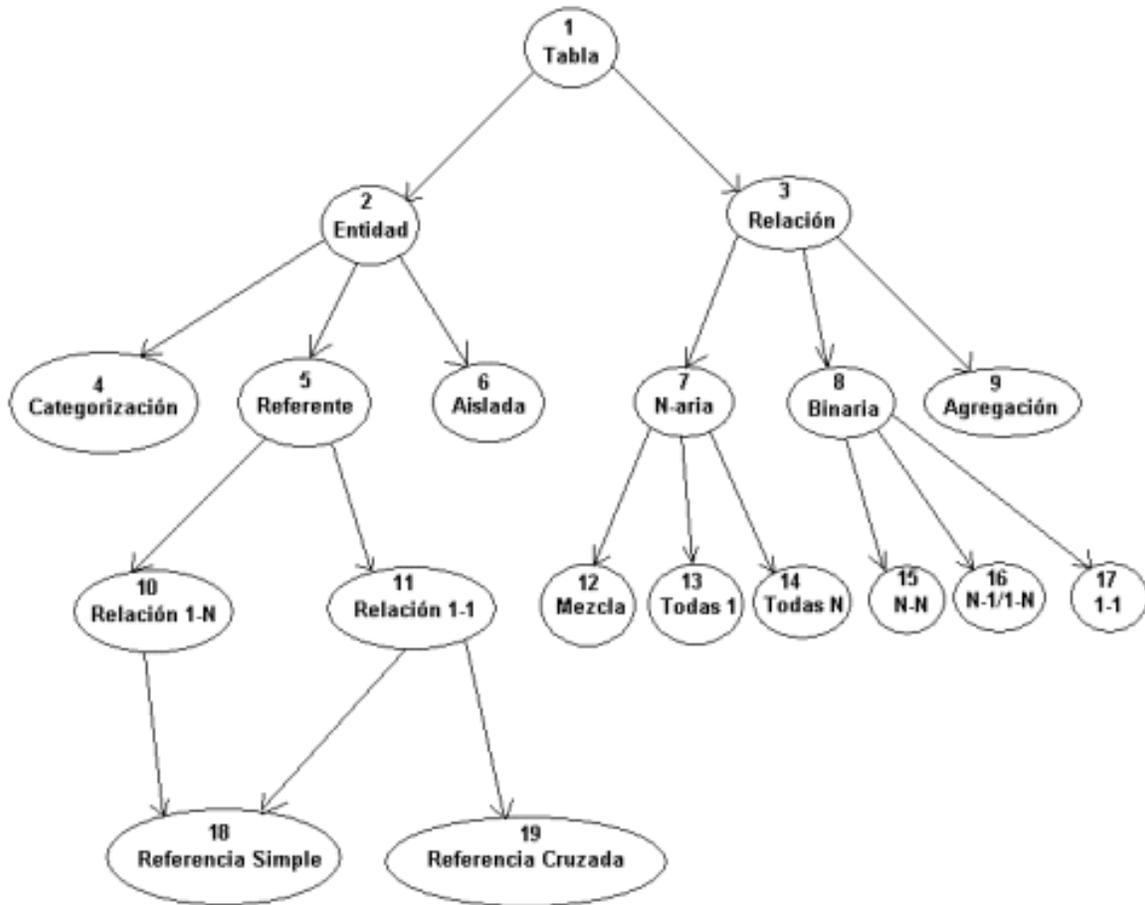


Figure 1

Figura 1. Diagrama de árbol

2 Determinar si una tabla corresponde a una entidad o a una relación

Lo primero que se debe realizar en el proceso del análisis es determinar si el modelo conceptual de una tabla corresponde a una entidad o a una relación.

Para realizar dicho análisis, se intentan probar distintos casos, mediante los cuales se podrá descartar las diferentes opciones.

2.1 Determinar si corresponde a una entidad aislada

Tal vez el término 'aislada' no es el más adecuado, debido a que en un modelo relacional bien hecho, muy difícilmente existan tablas completamente aisladas. En este análisis se refiere a entidades aisladas cuando una tabla no posee claves foráneas a otras tablas. Mediante el análisis de esta tabla no se puede saber a priori las relaciones en las que participa dicha tabla, pero sí se podrá determinar más adelante del análisis. Por lo tanto no es una entidad aislada, sino que más bien es una potencial entidad aislada, pero no se sabrá hasta finalizar el análisis de todas las tablas.

Determinar si una tabla corresponde a un entidad aislada es muy sencillo, lo único que se debe hacer es fijarse si dicha tabla posee claves foráneas. En el caso de que posea estamos seguros de que no es una entidad aislada y podemos proseguir con el análisis de la tabla, pero si se diera el caso que no posee ninguna clave foránea, entonces estamos seguros que corresponde a una entidad aislada, por lo que podemos agregar dicha tabla a nuestra estructura de almacenamiento entidades y pasar a analizar la siguiente tabla.

2.2 Determinar si corresponde a una categorización

Las categorizaciones se caracterizan por lo siguiente: toda la clave primaria de una tabla 'hija' forma una (y solo una) clave foránea a la tabla 'madre'.

Si se llega a este punto se sabe que la tabla posee por lo menos una clave foránea (por dicha razón no es considerada una entidad aislada). Lo primero que se debe hacer es fijarse si los atributos que componen a la clave primaria de la tabla componen a su vez una clave foránea. Con esto quiero decir que los atributos que componen la clave primaria NO componen a más de una clave foránea. En el caso que los atributos que componen la clave primaria no componen ninguna clave foránea, o que compongan a más de una clave foránea, se está seguro de que no nos encontramos frente a una categorización.

A continuación se plantea un ejemplo sencillo de categorización mediante el uso de tres tablas: empleados, gerentes y secretarias. La estructura física de las tres tablas es la siguiente:

Empleados	Gerentes	Secretarias
Número_Empleado (PK)	Número_Empleado (PKFK)	Número_Empleado(PKFK)

Table 1

El atributo Número_Empleado, tanto en la tabla Gerentes como en la tabla Secretarias, forma una clave foránea a la tabla Empleados.

Debido a que tanto la tabla Gerentes como la tabla Secretarias no poseen más claves foráneas, deducimos instantáneamente que no es una tabla que represente una relación, sino que existe una categorización. Por lo tanto, la representación sería la siguiente:

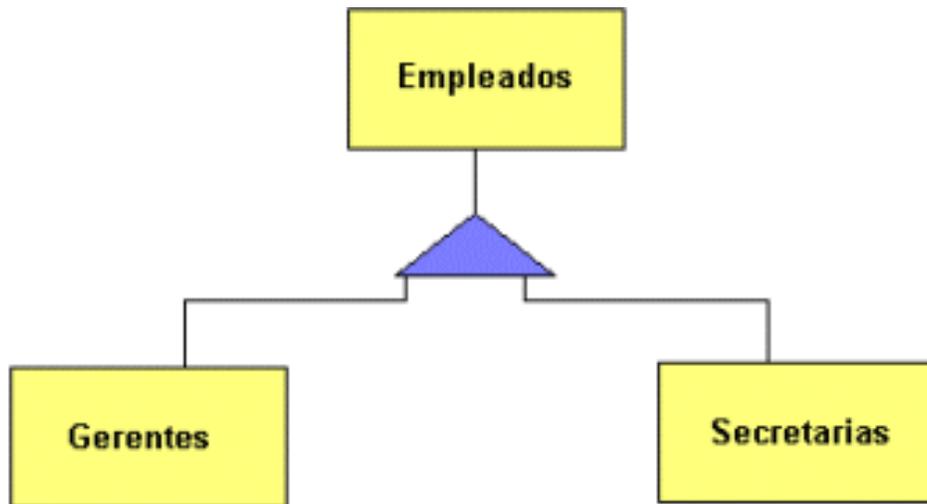


Figure 2

Imaginemos el siguiente caso:

Empleados	Gerentes	Secretarias
Número_Empleado (PK)	Número_Empleado (PKFK)	Número_Empleado(PKFK)Número_Computadora

Table 2

Si se nos diera este caso debemos realizar un análisis más profundo para poder determinar si la tabla Secretarias pertenece a una categorización, debido a que la representación conceptual de esta tabla podría ser cualquiera de las siguientes:

Caso 1:

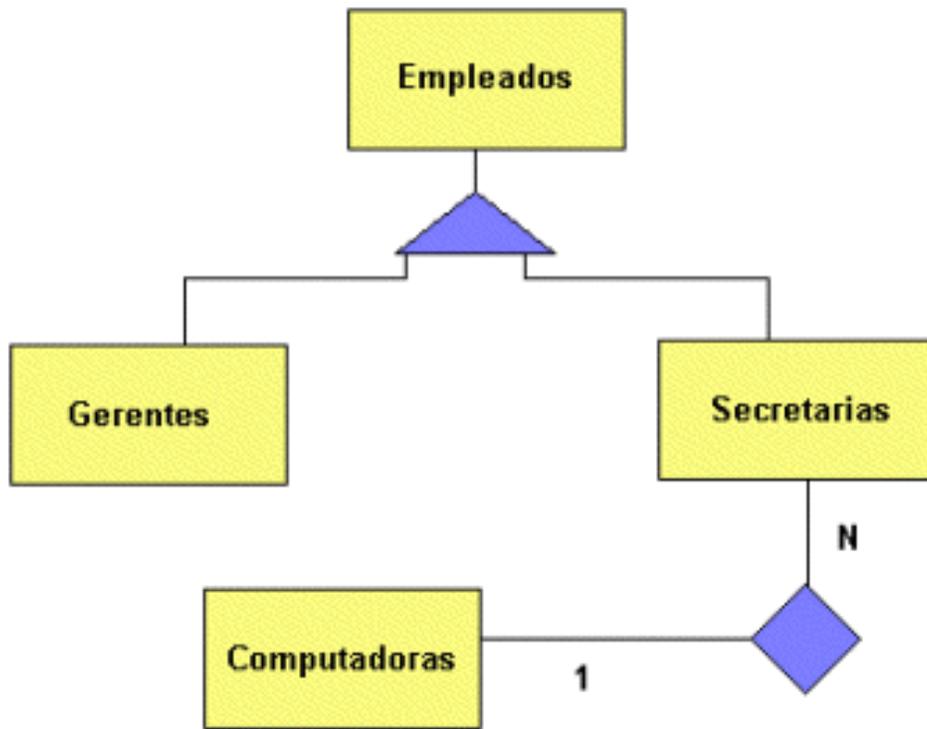


Figure 3

Caso 2:

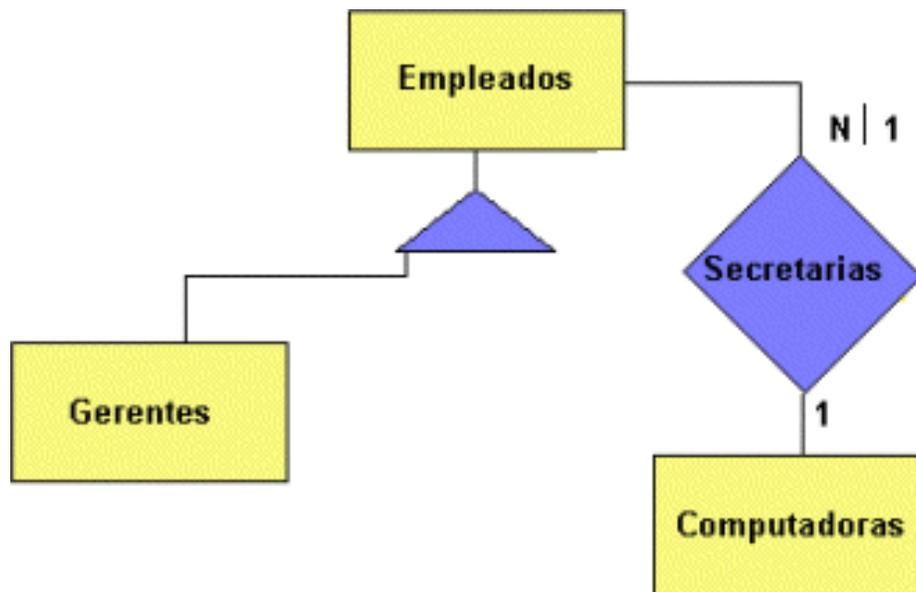


Figure 4

Como se puede observar, son representaciones completamente diferentes. En la primera Secretarias forma parte de una categorización, y en la segunda Secretarias es considerada una relación entre Empleados y Computadoras, con cardinalidades N-1 o 1-1.

En ocasiones es posible poder diferenciar entre los dos casos. La única forma de hacerlo es examinando el atributo Número_Computadora que pertenece a la tabla Secretarias y que hace referencia a la tabla Computadoras: si dicho atributo admite valores nulos, estamos completamente seguros que nos encontramos en el primer caso y no en el segundo. Esto es debido a que si Secretarias fuese una tabla que representa una relación entre Empleados y Computadoras, el atributo Número_Computadora NO podría aceptar valores nulos, debido a que por definición, las relaciones binarias son pares ordenados.

Si se nos plantea el caso de que el atributo Número_Computadora perteneciente a la tabla Secretarias no admite valores nulos, es imposible diferenciar entre los dos casos que planteamos anteriormente, por lo tanto debemos adoptar criterios para poder diferenciar entre ellos (por ejemplo, valores por omisión).

Una vez terminada esta parte del análisis sabemos si la tabla pertenece a una categorización o no, si perteneciera a una categorización se almacena en las estructuras de almacenamiento, y se analiza el resto de las claves foráneas que posee la tabla como si se tratase de una tabla que representa a una entidad referente.

2.3 Determinar si corresponde a una entidad referente

Entidad referente es una tabla que hace referencia a otras tablas (son las clásicas relaciones 1-N o 1-1).

Si llegamos a este punto sabemos que no nos encontramos frente a una tabla que representa a una entidad aislada y que tampoco corresponde a una categorización. Por lo tanto, ya se está seguro de que esta tabla es una tabla referente dado que tampoco puede representar una relación debido a que en una tabla que represente una relación los atributos que forman la clave primaria de la tabla deben formar también al menos una clave foránea.

Sabemos que cada clave foránea que posea la tabla representará una relación binaria (debido a que es el único tipo de relación que puede representarse sin utilizar una tabla) entre la tabla que nos encontramos

analizando y la tabla a la cual hace referencia la clave foránea. Además sabemos que la cardinalidad de dicha relación es 1-1, o bien 1-N, debido a que si fuese N-N se debería haber representado por medio de una tabla. Finalmente también sabemos que la cardinalidad correspondiente a la tabla que nos encontramos analizando es 1.

3 Análisis de una tabla que representa una relación

El análisis de una relación puede llegar a ser el más complejo debido a la cantidad de casos diferentes que existen (recuerde que una tabla podría representar una relación de N entidades, por lo tanto el número de tablas que podría llegar a relacionar es variable e infinito).

En todos los casos en los cuales una tabla representa una relación nos es imposible determinar las totalidades y parcialidades de la relación. Si se prueba que la tabla es una relación se coloca a esta junto con todos sus datos en nuestras estructuras de almacenamiento de relaciones (por ejemplo, nombre de la relación, tablas que relaciona, cardinalidad, etc.).

3.1 Relación binaria 1-1

Para explicar este caso plantearemos la siguiente situación: dada las entidades A y B que se relacionan mediante una relación con cardinalidad 1-1, tenemos que dado un elemento de A sólo existe un elemento de B y dado un elemento de B sólo existe un elemento de A. Ahora, para representar dicha situación mediante una tabla sólo existe una forma, y es la siguiente: una de las foráneas debe ser obligatoriamente la clave primaria (digo una porque recordemos que la clave primaria determina de forma única y mínima a cualquier tupla de la relación, y debido a que queremos representar una relación 1-1), con eso representaríamos una de las cardinalidades 1 (por ejemplo, la de A), pero aún nos falta representar la segunda cardinalidad 1 (siguiendo con el ejemplo la de B). Para realizar esto último debemos hacer uso de las claves candidatas, es decir, debemos hacer que la segunda clave foránea sea a su vez clave única (con esto representaríamos que B también posee clave única).

A continuación se plantea un ejemplo para intentar clarificar este punto.

Vehículos	Matrículas_Vehículos	Matrículas
Número_Vehículo(PK)	Número_Matricula(FK) Vehículo(PKFK)	Número_Matricula(PK)

Table 3

Obviamente la tabla Matrículas_Vehículos intenta representar una relación entre las entidades Vehículos y Matrículas. Como vemos, Número_Vehículo es una clave foránea y a su vez es clave primaria de la tabla, por lo que deducimos que la cardinalidad de Vehículos es 1. Ahora, si queremos representar una relación binaria 1-1 debemos hacer que los atributos que componen a la otra clave foránea (en este caso Número_Matricula) además de foráneos sean únicos en la tabla. Con esto último representaríamos que Matrículas también posee cardinalidad 1 en la relación.

A continuación presento un conjunto de tuplas para clarificar la necesidad de poseer la clave única.

Vehículos	Matrículas_Vehículos	Matrículas	Válido
8946	Num_Veh: 8946Num_Mat: SAB 555	SAB 555	Sí
8946	Num_Veh: 8946Num_Mat: SAK 430	SAK 430	No
1388	Num_Veh: 1388Num_Mat: SAK 430	SAK 430	No

Table 4

Como se ve en el ejemplo de tuplas anterior, existe una necesidad de especificar el atributo Número_Matricula como único.

En teoría deberíamos especificar las dos claves foráneas como únicas, pero debido a que la definición de clave primaria es que es única y no nula, queda implícito que si una clave foránea debe ser a su vez clave primaria, entonces dicha clave foránea también es única.

3.2 Relación binaria N-1 / 1-N

En este tipo de relación solo poseemos dos claves foráneas. Para esta situación, los atributos que componen la clave foránea correspondiente a la entidad que posee cardinalidad N deben formar a su vez la clave primaria de la tabla. A diferencia de el caso anterior, los atributos que forman la clave foránea correspondiente a la otra entidad NO pueden ser declarados como únicos.

3.3 Relación binaria N-N

A diferencia de los casos anteriores, este tipo de relación sí puede formar agregaciones, y debemos hacer ciertas consideraciones antes de comenzar su análisis.

Para representar este tipo de relación siempre se debe utilizar una tabla, y los atributos que compongan las claves foráneas correspondientes a las dos tablas que relaciona deben formar a su vez la clave primaria de la tabla.

En el caso que la clave primaria este formada por una sola clave foránea, y que a su vez no todos los atributos de dicha clave foránea formen a la clave primaria, podemos considerar que se quiere representar a una entidad no representada (es decir, que dicha entidad existe en el modelo conceptual, pero no en el físico y lógico).

A continuación se presenta un caso sencillo de este tipo de relación.

Alumnos	Asignaturas_Alumnos	Asignaturas
Número_Alumno(PK)	Número_Alumno(PKFK)	Número_Alumno(PKFK)
		Número_Alignatura(PK)

Table 5

Se observa que la clave primaria de la tabla Asignaturas_Alumnos se encuentra formada por dos claves foráneas. Debido a que la tabla Asignaturas_Alumnos no posee ninguna clave foránea a excepción de las que componen a la clave primaria, deducimos rápidamente que nos encontramos frente a una relación binaria N-N.

Por lo tanto, la representación conceptual de las tablas anteriores es la siguiente:

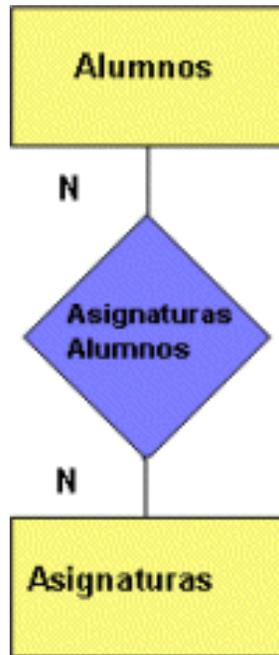


Figure 5

El caso de ejemplo que planteamos anteriormente ilustra un caso típico, en el cual podemos deducir sin ambigüedades la cardinalidad y tipo de relación existente, pero imagínese el siguiente caso:

Alumnos	Asignaturas_Alumnos
Número_Alumno(PK)	Número_Alumno (PKFK) Número_Asignatura (PKFK) Número_Semestre (FK)

Table 6

Asignaturas	Semestres
Número_Asignatura (PK)	Número_Semestre (PK)

Table 7

La tabla Asignaturas_Alumnos posee las siguientes características: la clave primaria de la tabla se compone por dos claves foráneas, pero además posee una clave foránea que no compone a la primaria. Es obvio que nos encontramos frente a una tabla que representa una relación, pero el problema es determinar el tipo de relación existente. Si nos enfrentamos a un caso similar a este, se nos pueden plantear dos posibilidades que mostramos a continuación:

Caso 1:

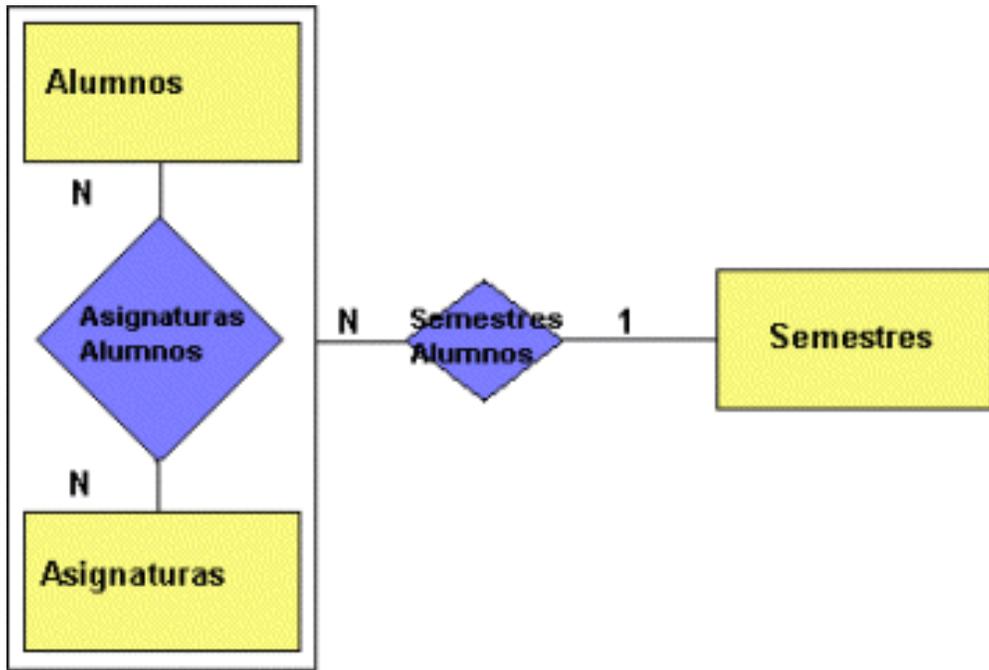


Figure 6

Caso 2:

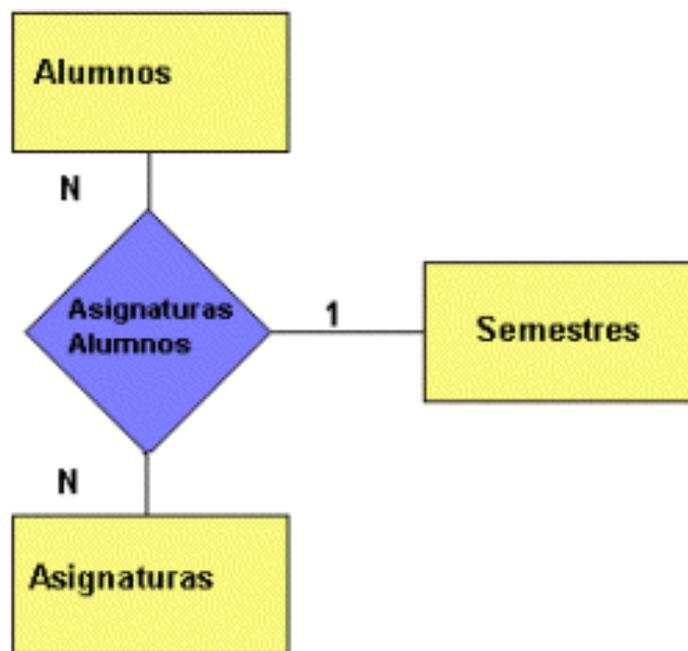


Figure 7

Como notará, existe una gran diferencia entre las dos posibilidades, debido a que la primera corresponde a una relación binaria formando una agregación, pero en el segundo caso la relación es ternaria. Esta situación se plantea debido a que si una relación forma una agregación que se relaciona con otra entidad, si dicha relación (entre la agregación y la entidad) posee cardinalidades 1-1 o N-1, existe la posibilidad de que no se represente la relación por medio de una tabla.

Para poder resolver este problema sólo tenemos dos posibilidades. La primera es examinando los atributos que forman la clave foránea que no compone la clave primaria de la tabla que representa la relación y en el caso de que dichos atributos admitan valores nulos, entonces deducimos directamente que nos encontramos frente al caso de una agregación, debido a que si fuese una relación ternaria no debería admitir valores nulos. Si la primera opción falla, tenemos una última opción y es intentar probar una relación 1-1 cruzada entre la relación y la entidad (en nuestro ejemplo entre la tabla *Asignaturas_Alumnos* y *Semestres*) en el caso de que probemos que existe una relación 1-1 cruzada podemos decir con total seguridad que nos encontramos frente a una agregación.

3.4 Análisis de una relación n-aria

Si llegamos a este punto, sabemos con certeza de que la tabla representa una relación, y que a su vez esta relación no es binaria. Por ende, nos queda sólo determinar si es o no una agregación, y en el caso que no lo sea analizar las características de la relación (cardinalidad, entidades que relaciona, etc.).

Para realizar este análisis consideraremos que una relación n-aria es una relación que relaciona N entidades, siendo N un número mayor que dos (esto lo haremos porque las binarias las analizamos en la sección anterior, pero no se debe perder de vista que una relación binaria es un caso particular de una relación n-aria). A pesar de este hecho, existen tres posibilidades bien diferenciadas en una relación n-aria, vinculadas con sus cardinalidades: la primera es que todas sus cardinalidades sean uno, la segunda es que todas sus

cardinalidades sean N y finalmente que sus cardinalidades sean una mezcla entre unos y enes.

3.5 Relación n-aria con todas sus cardinalidades N

Si una tabla representa una relación n-aria con todas sus cardinalidades N, entonces sabemos con certeza de que dicha tabla no posee claves únicas para representar su cardinalidad (debido a que no necesita esto).

Distinguir este tipo de relación es sumamente fácil, debido a que para representar esta cardinalidad todas las claves foráneas que forman la relación deben formar a su vez la clave primaria de la tabla.

Ahora puede darse el caso de que se quiera representar una relación entre varias entidades, en donde una de estas entidades no se represente tanto en el modelo lógico como en el físico. No tenemos duda que es una relación n-aria con todas sus cardinalidades N, debido a que no posee claves únicas y todas las claves foráneas forman la clave primaria. Pero no todos los atributos que componen a la clave primaria son foráneos, por lo tanto deducimos que existen entidades no representadas en el modelo físico. Aquí se evaluarán dichos atributos según los criterios que nosotros impongamos (por ejemplo, cada atributo es una entidad no representada, preguntar al usuario, etc.).

A continuación planteo un ejemplo, de una relación n-aria con entidades no representadas:

Tipos_de_Movimientos	Conformes	Tipos_Conformes
Número_Tipo (PK)	Número_Conforme (PK)	Número_Tipo (PKFK) Número_Conforme (PKFK) Fecha (PK)

Table 8

En las tablas anteriores, notamos que en la tabla Tipos_Conformes el atributo Fecha forma parte de la clave primaria de la tabla, pero no compone ninguna clave foránea, por lo tanto deducimos que es una entidad no representada (no sería lógico tener en el modelo físico una tabla con todas las fechas posibles; en cambio, en el modelo conceptual sí es útil y muchas veces necesario).

La representación conceptual de la tabla anteriormente expuesta es la siguiente:

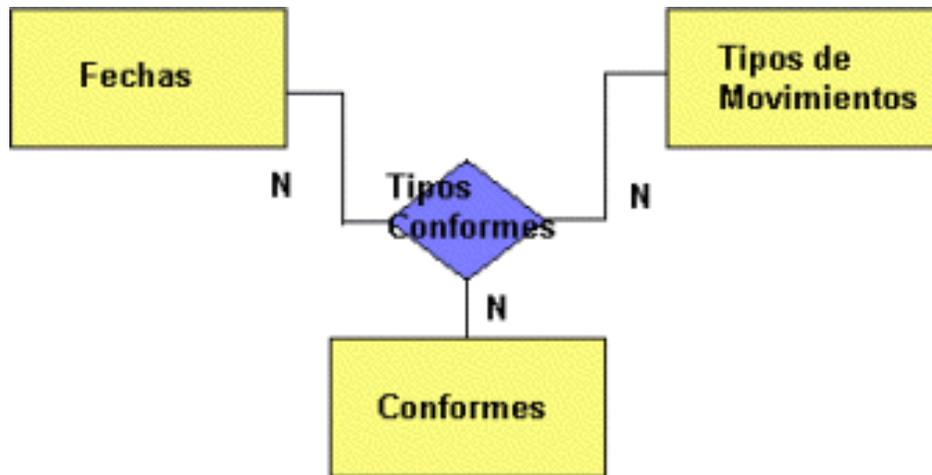


Figure 8

3.6 Relación n-aria con todas sus cardinalidades 1

Para determinar si una relación pertenece a este tipo debemos estudiar sus claves candidatas (es decir, sus claves únicas). En este caso sabemos que toda clave foránea que pertenezca a una entidad que esta tabla relaciona se debe encontrar ya sea en la clave primaria o en alguna de sus claves únicas.

A continuación planteamos un ejemplo de este tipo de relación por medio de una relación ternaria:

Matrículas	Vehículos
Número_Matrícula (PK)	Número_Matrícula (PK)

Table 9

Departamentos	Matrículas_Vehículos_Departamentos
Número_Departamento (PK)	Número_Matrícula (PKFK)Número_Vehículo (PKFK)Número_Departamento (FK)

Table 10

En este caso existirán dos claves únicas además de la clave primaria. Dichas claves únicas son (Número_Matrícula, Número_Departamento) y (Número_Vehículo, Número_Departamento).

Entonces en este caso deducimos que la relación es 1-1-1 debido a que al haber una clave foránea que no es primaria, entonces sabemos que la cardinalidad de la entidad a la cual hace referencia dicha clave foránea es 1. Ahora listamos todas las claves únicas con los atributos que la componen. Para cada caso aquel atributo que no se encuentre en una clave única y que nosotros sepamos que hace referencia a una entidad que relaciona la tabla, entonces sabemos con certeza que tiene cardinalidad 1, es decir, si Número_Matrícula-Número_Departamento componen una clave única sabemos que Número_Vehículo tiene cardinalidad 1.

Ahora pasaremos a justificar lo anterior con un ejemplo, ingresando algunas tuplas a las tablas de la relación antes citada.

Matrículas_Vehículos_Departamentos			Válido
Núm_Mat	Núm_Veh	Núm_Dep	
SAK 445	4670	10	Sí
SAK 445	890	10	No
SSD 320	430	11	Sí
DFD 440	4670	10	No

Table 11

Debido a que Número_Matrícula, Número_Vehículo son clave primaria de la tabla, entonces deducimos que la entidad Departamentos tiene cardinalidad 1.

Ahora, a partir del ejemplo que mostramos ingresando tuplas, deducimos que un valor de Número_Vehículo y Número_Departamento estos sólo pueden aparecer juntos en una misma tupla una sola vez en toda la tabla. Por ende estos dos atributos forman una clave única, deduciendo entonces que la entidad Matrículas tiene cardinalidad 1. De la misma forma ocurre con la entidad Vehículos.

Por lo tanto la representación conceptual de este conjunto de tablas es la siguiente:

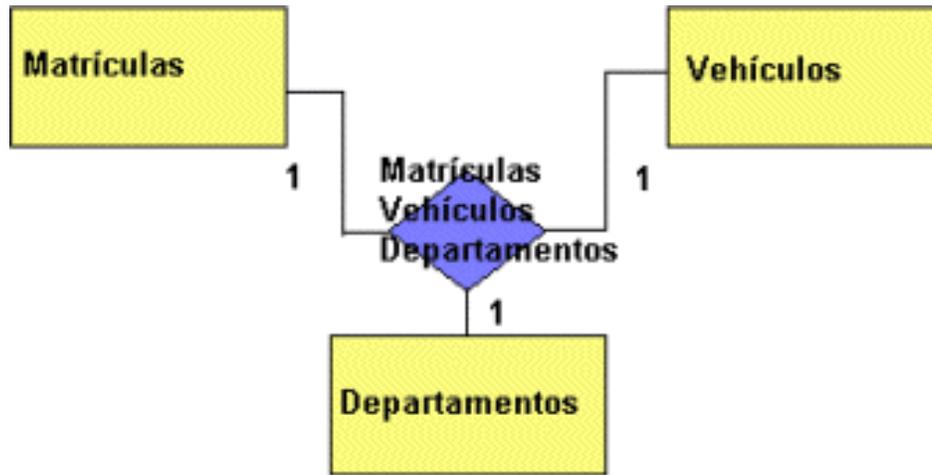


Figure 9

3.7 Relación n-aria con sus cardinalidades mezcladas

Se entiende por cardinalidad mezclada la cardinalidad de la relación no es ni todas uno ni todas enes, sino que la cantidad de cardinalidades unos y enes son variables.

Para resolver este tipo de relación, nuevamente haremos uso de las claves candidatas (claves únicas). El análisis lo haré basándome en un ejemplo.

Personas	Personas_Garantes
Número_Persona (PK)	Número_Persona_Garante (PK)

Table 12

Conformes	Conformes_Personas
Número_Conforme (PK)	Número_Persona (PKFK)Número_Conforme (PKFK)Número_Persona_Garante (FK)

Table 13

Como podemos observar en los esquemas que describimos anteriormente, la tabla que representa la relación tiene la siguiente clave primaria compuesta: Número_Persona, Número_Conforme; por lo cual deducimos directamente que la cardinalidad de la entidad Personas_Garantes es 1.

Luego tras analizar las claves únicas que posee la tabla deducimos que posee una clave única compuesta por los siguientes atributos: Número_Conforme, Número_Persona_Garante, por lo cual deducimos que la entidad Personas también posee cardinalidad 1. Finalmente al no poseer más claves únicas llegamos a la conclusión de que la cardinalidad de esta relación es 1-1-N.

4 Conclusión del ejemplo

Se ha podido observar todo el análisis exhaustivo que se ha realizado a través del código fuente (diseño físico), se puede obtener el diseño lógico aplicando Ingeniería Inversa y también el Diseño Conceptual.