



TEMA II

LA BASE DE DATOS RELACIONAL

INTRODUCCIÓN

Muchos de los conceptos que se expondrán a lo largo de este tema y subsiguientes ya han sido tratados por lo que no serán redefinidos pero sí se presentará una visión desde otra óptica.

Los datos, son una representación de una pequeña porción del mundo real, una población. Una base de datos, representa varias poblaciones de interés a una familia de aplicaciones. Estas poblaciones pueden ser distintas y diversas, pero relacionadas de algún modo. La base de datos debe reflejar dichas relaciones que existen dentro de la población si va a ser útil. En esencia, el significado de esas poblaciones es el que se corresponde al concepto de ENTIDAD, y las asociaciones entre las poblaciones corresponden al de RELACIÓN.

Si se considera por ejemplo, un negocio de venta de cualquier tipo de mercadería, puede observarse que los componentes del sistema operativo que lo sostiene, consiste por lo menos en las siguientes poblaciones o entidades:

- Las mercaderías
- Los vendedores que venden dichas mercaderías
- Las órdenes de compra de mercaderías
- Las facturas de compra de la mercadería
- La cuenta del vendedor por las mercaderías entregadas
- Los cheques de pago de los clientes
- Los clientes con cuenta corriente
- etc.

Los ejemplos citados más arriba representan poblaciones. Los datos de estas poblaciones son las ocurrencias o instancias de las entidades y conformarán “la base de datos”. Además se agregan los vínculos que existen entre ellas – las relaciones – como ser entre:

- Clientes y Órdenes de compra
- Órdenes de compra y Mercaderías
- Órdenes de compra y Vendedores
- etc.

Las mismas deben ser convenientemente previstas en los contenedores que almacenarán esos datos. Dado que las relaciones son de uso corriente en la arquitectura, es de interés conocer más acerca de ellas. Por lo tanto se dedicará este tema al examen de las relaciones y a su aparición en las bases de datos.

La relación describe una forma en que pueden asociarse individuos. Una relación es una simple asociación entre exactamente dos o más individuos. Por lo general, se vinculan individuos de poblaciones distintas, pero, aunque no parezca, más aún son utilizadas por los DBMS las relaciones entre individuos pertenecientes a una misma población.

RELACIONES SIMPLES

El triplete o formas de relación

La forma o concepto de relación, incluye tres elementos: dos individuos y una forma en que los individuos se afectan cada uno al otro conocida con el nombre de **relación**. El **triplete es la forma de relación**.

- El **sujeto** es el individuo del cual parte la relación.
- La **relación** se caracteriza a menudo por uno de los verbos activos o pasivos que muestran cómo pueden interactuar los individuos entre sí.
- El individuo que recibe la acción, es el **objeto** de la forma de relación.

Para dar forma a una relación, la simbología utilizada será:

$$a R b$$

Donde la primera letra **a**, representa el sujeto; la segunda letra **R**, expresa la relación, y **b** es el objeto. Por ejemplo, la siguiente relación:

$$g > s$$

Expresa que la relación **mayor que**, simbolizada por $>$. La misma se puede aplicar entre dos números: **g**, el número de la izquierda, es uno del cual sostiene que es mayor que **s**, el número de la derecha. Debe notarse que en esta instancia, tanto el sujeto como el objeto son el mismo tipo de **entidad**.

Alcance y Dominio

El dominio es una población de la cual se extrae el sujeto. El alcance es la población de la cual se extrae el objeto. Para la relación expresada anteriormente, el dominio y el alcance, son ambos los números reales. El problema es que una vez que se ha seleccionado un número **n** para el dominio, se estará restringido con respecto a los números que pueden ubicarse a la derecha. Puede escribirse esto simbólicamente como:

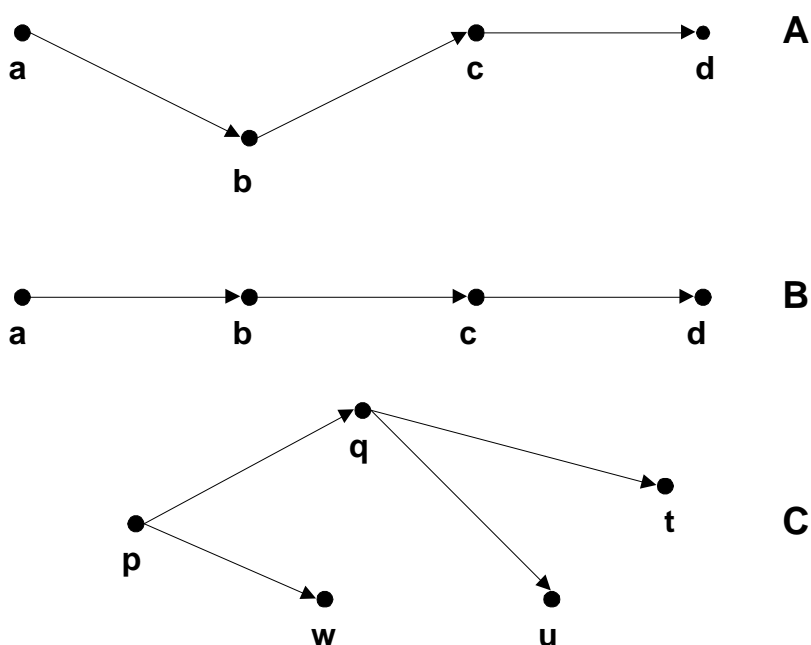
$$n > n'$$

Donde **n** es un conjunto de números reales, y **n'** es un conjunto truncado - el de aquellos entre los cuales se puede seleccionar **n'** de modo tal de no negar esta forma de relación -.

Grafos

Un modo de expresar un conjunto de formas de relación, es mediante un grafo. Un **grafo** es una figura geométrica que consiste solamente en vértices y arcos, o segmentos de recta. Un vértice (o punto) puede representar a un **individuo** que participa en una forma de relación; el individuo puede ser el **sujeto** o el **objeto** de dicha forma de relación. Un arco es un segmento de línea (no necesariamente recta) dirigido mediante una punta de flecha, que representa una **relación**. Cualquier arco de un grafo válido, debe conectarse con **dos** vértices. Es obvio que una forma de relación debe tener **dos** elementos para relacionar, **el sujeto y el objeto**.

En la gráfica siguiente se muestran tres formas de relación. Cada arco relaciona un par entre cuatro individuos **a, b, c y d**.



Las tres formas de relación son:

$$a R b; \quad b R c; \quad c R d$$

Para **A** y **B**. Este conjunto de relaciones, es lineal. Esto es, se puede expresar con un segmento de línea con cuatro vértices en el mismo. Como ejemplo de lo que esto significa, se puede suponer que **R** es una relación **es Padre de**. En este caso, **a** es padre de **b**, **b** es padre de **c** y **c** es padre de **d**. Resulta interesante poner en claro que esta relación no determina que **a** sea padre de **c**; **a** es abuelo de **c**. Si se desea expresar esta clase de disposición, **R** puede definirse como **es Antepasado de** en lugar de **es Padre de**. Se puede observar en el caso **C** que este **no es un conjunto lineal** de relaciones:

$$p R q; \quad q R t; \quad p R w; \quad q R u$$

Propiedades de la Relaciones

Reflexividad

Se dice que una relación es **reflexiva** cuando puede aplicarse con validez a un elemento de sí mismo. Es decir, el mismo elemento resulta ser a la vez sujeto y objeto de una forma de relación. En forma simbólica se tiene:

$$a R a$$

Si se retoma el ejemplo de la relación **es mayor que** o la relación **es padre de**, se puede hallar que no son reflexivas. Resulta claro que un número no puede ser mayor que sí mismo, y que una persona no puede ser a su vez padre de sí mismo.

Simetría

Una relación es simétrica cuando es **siempre** posible intercambiar el **objeto** y el **sujeto** en la forma de relación y obtener otra forma de relación verdadera. Esto se define mediante:

$$a R b \equiv b R a$$

Donde \equiv significa equivalente a. En consecuencia, la simetría significa que la relación tiene una cualidad tipo espejo, vale decir que objeto y sujeto pueden intercambiarse.

Evidentemente, ninguna de las relaciones anteriores, **es mayor que** y **es padre de**, son simétricas, sino que son asimétricas.

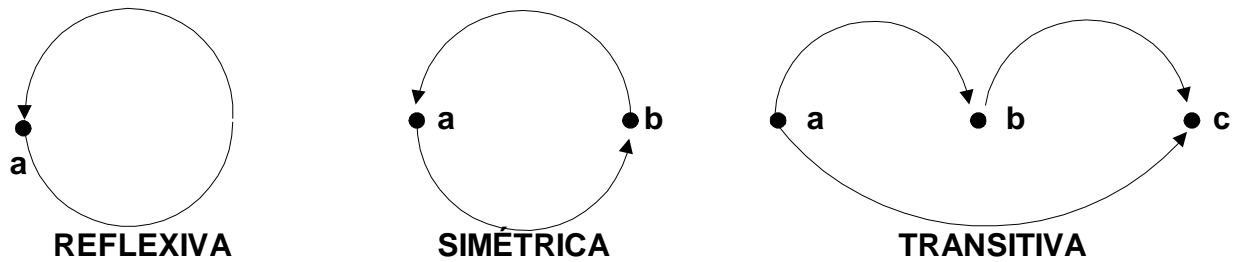
Transitividad

Una relación es transitiva cuando se aplica lo siguiente:

$$a R b \text{ y } b R c \Rightarrow a R c$$

Se tienen aquí, dos formas de relación, donde **b** es el objeto de la primera y el sujeto de la segunda. Cuando un elemento **b** es el objeto de una relación y el sujeto de la segunda, y la relación **R** es *transitiva*, entonces el sujeto **a** de la primera, y el objeto **c** de la segunda, forman un par ordenado para el cual se aplica la relación.

A continuación, se mostrarán gráficamente las propiedades enunciadas:



MULTIPLICIDAD DE RELACIONES

La multiplicidad describe el número de sujetos y objetos que participan en formas de relación dentro de una población. Esto es, dado un sujeto, la forma de expresión de cuántos se relacionan con él. Dicho de otra forma, se trata del concepto de funcionalidad de una relación. La multiplicidad es una característica de la forma de relación R .

Existen exactamente tres alternativas que ya se mencionaron, y son **1 a 1**, **1 a n** y **n a m**. Se considerará ahora otro tipo denominado:

Forma de Relación Inversa

En la lista original de tipos de relación, la relación muchos a uno, no ha aparecido. La razón de esto es que, dada una relación, no resulta difícil crear otra relación R^* , la cual es su inversa. La *relación inversa* intercambia el sujeto y el objeto como se indica:

$$a R b \equiv b R^* a$$

La inversa de la relación **I**, “es emitido por”, es **I*** que significa “emite”. El sujeto y el objeto se han invertido, en consecuencia, por ejemplo, en lugar de tener un cheque emitido sobre una cuenta, se puede hablar ahora de una cuenta que recibe un cheque. Estos dos, son totalmente equivalentes:

$$c I a \equiv a I^* c$$

Puede verse de inmediato que si la relación a invertir es **uno a uno**, **luego la relación inversa también será uno a uno**. Sin embargo, si la relación original era **uno a muchos**, entonces la relación inversa será **muchos a uno**. Finalmente se notará el hecho trivial de que la inversa de la inversa es la relación original:

$$(R^*)^* \equiv R$$

ORDEN EN RELACIONES, CLAVES, ARCHIVOS CLASIFICADOS Y LISTAS DE CADENAS

Orden

Se han discutido los tipos y categorías de relaciones que existen entre individuos en una población. El tipo más importante de relación es el que se refiere a *la posición de los individuos respecto de los demás*. Este tipo de relación se denomina **relación de orden**. Las relaciones de orden pueden ser parciales o completas. Las relaciones de orden completas establecen una posición para todo individuo de la población; las relaciones de orden parcial tendrán un ordenamiento para algunos individuos, y sólo un ordenamiento relativo para otros.

Precede

Una relación de orden típica es **precede**, para la cual se usará el símbolo **P**. Se dice simbólicamente que el elemento **a**, precede al elemento **b**; en consecuencia, la relación **P** indica la posición de cualquier elemento en el archivo, suponiendo que se ha acordado de qué forma cada elemento se ubica respecto a su vecino. Se tiene entonces:

$$a \ P \ b$$

Examinando ahora las cualidades de **P**, se tiene que:

- **No es reflexivo** - un elemento no puede precederse a sí mismo.
- **Es asimétrico** - si un elemento precede a otro, el segundo elemento no precede al primero.
- **Es transitiva** - si **a** está antes que **b**, y éste está antes que **c**, entonces **a** precede a **c**.

Sigue

La inversa de **precede** es **sigue**. Se tiene entonces que:

$$b \ P^* \ a$$

donde **P*** significa **sigue**.

Claves y Orden

Para la mayor parte de las estructuras de datos, se requiere que un registro o segmento, tenga un atributo (o combinación de atributos) que posea un único valor. Este elemento es la clave que **identifica** a un individuo que es descripto por el registro. De otro modo, no se podría relacionar ese registro con algún individuo de la población. Por lo general, al menos un atributo cumple con este requisito. Se asumirá de aquí en adelante, que **un único atributo identifica al registro y se denominará clave primaria**. Todo aquello que se diga acerca de la clave primaria, se aplica también a las claves compuestas. Volviendo a conceptos vistos anteriormente, de acuerdo con la primera forma normal, aplicada tanto a entidades como a relaciones, *“toda entidad debe tener al menos una clave...”*, *“toda relación tiene por clave, la concatenación de las claves de las entidades que forman parte de su colección...”*. Llevado al entorno de trabajo de

implementación física, cada entidad se transformará en un fichero, y eventualmente, las relaciones también lo harán, dependiendo de su funcionalidad.

Claves para Determinar el Orden

La clave relaciona los datos acerca de un individuo con los demás individuos de la población. Puede también ser utilizada para determinar la secuencia de registros en el archivo. Para obtener esto, se exigirá un ordenamiento sobre la clave y sobre el código que la reemplace en el medio físico, de modo tal que, dadas dos claves, se puede determinar cuál es la mayor. Esto, a su vez, determinará cuál de ellas precede a la otra en el archivo. Se puede utilizar otro criterio para determinar el orden. Por ejemplo, para un archivo de transacciones (procesos de alta, baja y modificación), pueden ser ubicados en el archivo de log, de acuerdo con su fecha y hora de llegada. Esto puede cumplirse independientemente de la clave o tipo de transacción. Aun si la transacción no contiene clave, el tiempo de arribo puede ser un criterio implícito - esto es, puede no formar parte de los datos incluidos en dicho registro, sino agregado al momento de hacer la inserción -.

Operaciones

Para representar las propiedades de un archivo ordenado, se propondrán dos operadores. El primero de ellos, **clave**, extrae la clave de un registro o segmento; el segundo, **loc**, encuentra la ubicación física de dicho registro o segmento dentro del archivo. Como ejemplo del uso de **clave**, se puede decir que la clave de registro **b** es mayor que la clave de registro **a** como se indica:

$$\text{clave (b)} > \text{clave (a)}$$

Archivo Ordenado

Se dice que un archivo de registros o segmentos es un archivo ordenado, cuando la posición del registro o segmento en el archivo *es definida por su clave*. Si se encuentra con registros de acuerdo con claves crecientes, de menor a mayor, se define al archivo ordenado como de **orden Ascendente**. Si a medida que el archivo es recorrido, y se encuentra con los registros de acuerdo con claves descendentes, el archivo ordenado, estará en **orden Descendente**. En el primer caso se tendrá:

$$\text{clave (b)} > \text{clave (a)} \quad \text{vale decir que} \quad \mathbf{a P b}$$

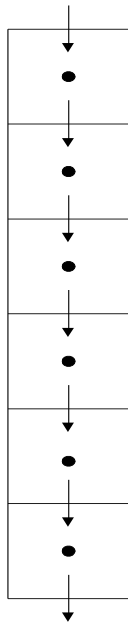
Donde **P** significa “precede”.

Existen dos formas de determinar el modo de recorrer un archivo para encontrar los registros en la secuencia apropiada:

- **ubicación física del registro**
- **algunas otras técnicas como punteros (listas, enlaces, dobles enlaces, etc.)**

En la siguiente página, se discuten y grafican las posibilidades que brindan ambos, destacando las ventajas de uno sobre otro.

Archivo Clasificado



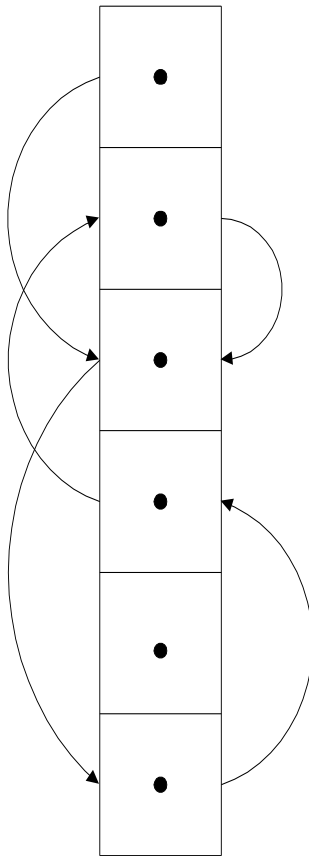
Cuando la forma en que se recorre un archivo está determinada por la ubicación física del registro en el archivo, se lo puede llamar **archivo clasificado**. La clave del registro provee un sistema completo de ordenamiento. Esta técnica, puede verse expresada gráficamente en la figura mostrada a la izquierda. Nótese que:

- un registro es designado por un punto
- una ubicación física en el archivo es señalada como un cuadrado
- la relación **precede**, con respecto a la clave, es designada con un arco (segmento orientado)
- la ubicación de los registros, adyacentes en la figura, es también de adyacencia en el archivo

Para un archivo clasificado en orden ascendente, el registro con la clave menor se puede hallar en la ubicación con la dirección menor. Esto se representa simbólicamente:

clave (b) > clave (a) vale decir que loc (b) > loc (a)

Lista con punteros



Un archivo clasificado presenta la ventaja de que se conoce la posición relativa de cada registro. Pero tiene la desventaja de que la adición o eliminación de registros, altera la ubicación física de un gran número de otros registros. La lista, elimina esta última dificultad proveyendo un puntero en cada registro hacia su sucesor. Esto permite liberarse de la obligación de tener adyacencia física entre registros sucesivos.

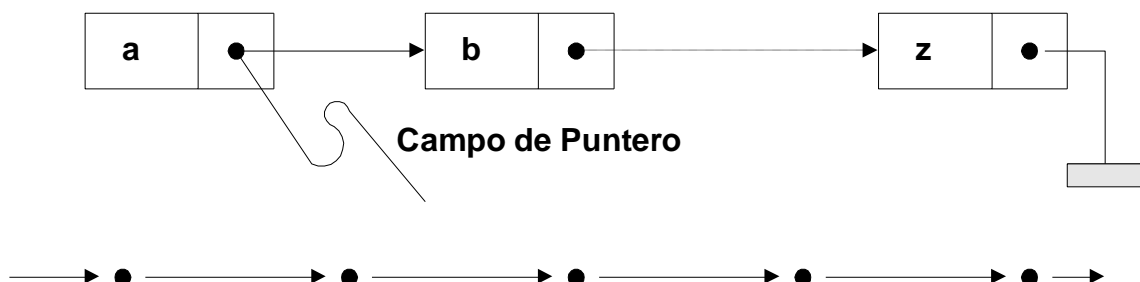
Para señalar la existencia del símbolo puntero, se usará el operador **punto**. Dicho operador ubica el puntero del registro, que tiene la ubicación física del registro inmediato siguiente, de modo que se obtiene:

$$a P b: \text{ punto } (a) = \text{ loc } (b)$$

Esto es, ya que el registro **a** precede al **b**, y se espera encontrar un puntero en el registro **a**, con la ubicación física del registro **b**.

La estructura de un archivo ordenado bajo la forma de una lista puede observarse en la figura de la izquierda. Aquí los registros de posiciones adyacentes no guardan ninguna relación de orden entre sí. En lugar de ello, dicha relación, está representada por medio de los punteros, representados por los arcos en la figura.

Esto puede apreciarse en la figura siguiente donde se observa que es requerido un campo adicional para apuntar desde un registro a su sucesor. Obsérvese que el *último* registro, tiene también un puntero, pero éste es una señal terminal - aquí una bajada a tierra - para designar a este registro como último. Ocurre algo similar para el primer registro de la lista, al cual, necesariamente debe apuntar una **cabecera**.



Un archivo ordenado, ya sea que esté físicamente ordenado o a través de punteros, se representa simplemente como una línea dirigida con los vértices representando registros del archivo, tal como se ve en el gráfico superior.

Ventajas

La facilidad de agregar o eliminar elementos de una lista en cadena, la hace atractiva. Las técnicas para llevar a cabo esto, ya fueron presentadas en otra materia. La importancia de la uti-

lización de punteros yace en la posibilidad de relacionar individuos de diferentes poblaciones (entidades). Por otra parte, proveen un medio de sobreponer múltiples relaciones sobre una sola población. Como desventaja, podría indicarse fundamentalmente, que las direcciones de ubicación física de cada registro requieren como soporte, la memoria del equipo.

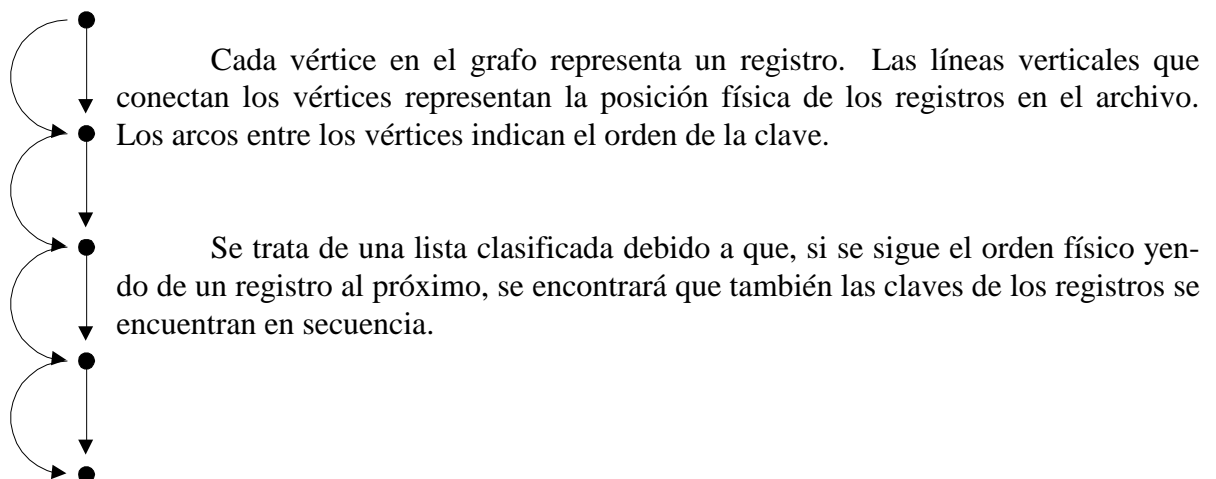
GRAFOS COLOREADOS

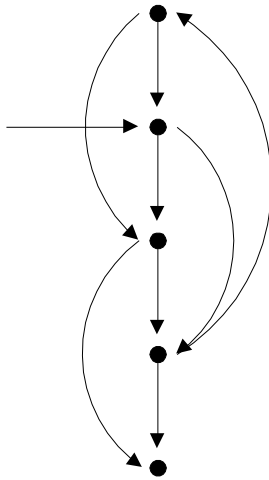
Un grafo consiste en vértices y arcos: un vértice representa un individuo; un arco representa una relación entre dos individuos. Una población es un conjunto de individuos y está representada en un grafo mediante un conjunto de vértices. En los sistemas reales, normalmente puede existir más de una relación entre individuos en una población. El problema que se presenta ahora, es determinar si existe alguna forma de representar más de un tipo de relación en el mismo grafo.

Se verá a continuación el caso del grafo **multi-relacional**. Nuevamente, consiste éste en un número de vértices, cada uno representando a un individuo. Suponga ahora que existen dos clases de relaciones y que deben ser representadas ambas. Esto significa que se necesitan dos tipos de arcos. Para distinguirlos entre ellos, se pueden colorear o bien, representarlos con distintos tipos de líneas. Líneas finas, gruesas, de trazos, de puntos, con círculos encima, etc. Cada línea distinta, representa una relación diferente dentro de la misma población.

Ejemplo

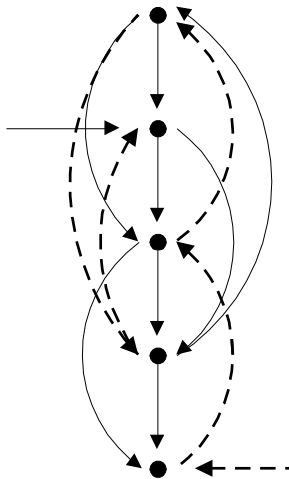
En un ejemplo anterior, ya se había utilizado un grafo coloreado sin definir su significado. Se presenta ahora un ejemplo en la siguiente figura y representa una lista clasificada.





En este gráfico aparece un archivo ordenado, pero no un archivo clasificado. Los registros aparecen en el orden físico según las líneas verticales entre cada vértice, el cual representa un registro. Las líneas curvas también conectan los vértices. Estas representan un orden de clave ascendente. Para encontrar los registros en orden de clave ascendente, se debe seguir dichos arcos. Estos arcos también representan punteros cuando se trata de una lista. No tiene mayor sentido utilizar dos conjuntos de arcos con diferentes colores o tipos de líneas para los punteros y la secuencia de claves: ambos datos, son idénticos.

Si se desea imponer un orden descendente al mismo archivo, se deben recorrer los datos en el orden opuesto:

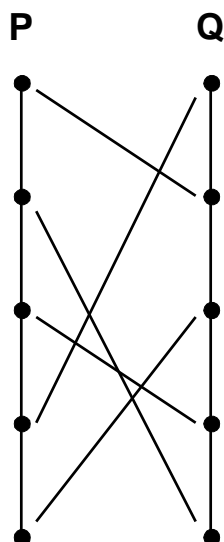


El orden descendente, está indicado por los arcos de líneas llenas, y el orden descendente, por los arcos con líneas de trazos. Dado que los órdenes ascendentes y descendentes son exactamente opuestos, un arco de línea llena en una dirección aparece como un arco de línea de trazos entre los mismos dos vértices, pero en dirección opuesta, y el grafo es simétrico salvo en lo que se refiere a los puntos de entrada o cabecera.

Poblaciones Diferentes

Relaciones uno a uno

Una relación que existe dentro de una población puede ser de un sólo tipo, mientras que la relación existente entre miembros de diferentes poblaciones, puede ser de **naturaleza distinta**. Considérense las poblaciones **P** y **Q** tal cual como se muestra en la siguiente figura como líneas verticales:

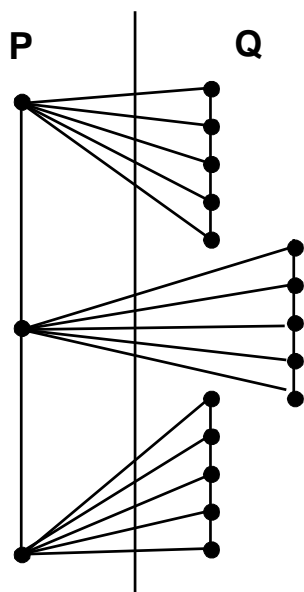


Estas líneas verticales, representan organizaciones dentro de **P** y **Q**. Supóngase que existe una relación entre un individuo en **P** y uno de **Q**. Por ejemplo, **P** pueden personas y **Q** pueden ser alumnos, de esta manera, habrá una **relación 1 - 1** entre ellos (un alumno es una persona y una persona puede ser alumno). Un arco, conecta un individuo en **P** con exactamente un individuo de **Q** para representar este caso.

Para implementar dicha disposición, se necesita como mínimo un conjunto adicional de punteros en los segmentos de la población **P**. Estos punteros, darán la ubicación física del correspondiente individuo de **Q**.

Relaciones uno - muchos

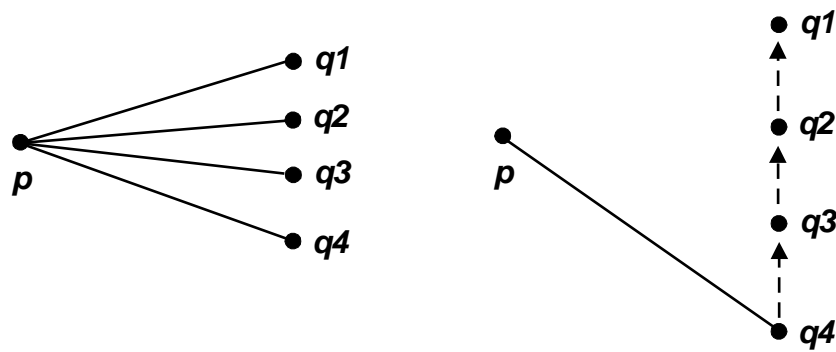
Considérense una población **P** y otra **Q** con una relación entre un miembro de **P** y varios miembros de **Q**. La relación uno - muchos, se presenta solamente en una dirección: no se trata de una relación muchos - muchos. Un miembro de **Q** se relaciona con uno sólo de **P**.



Considérese en primer lugar las dos poblaciones solamente con respecto a la relación que interesa, sin tener en cuenta cualquier organización que pueda existir dentro de cada población. La situación se puede ver en la figura de la izquierda, donde los miembros de **P** aparecen como vértices a la izquierda de la línea vertical y los miembros de **Q** aparecen como vértices a la derecha de la vertical. Este es el primer aspecto: la relación uno a muchos se representa por un **sub-árbol** simple de dos niveles con una sola raíz en **P** y varias hojas en **Q**. *Existe un sub-árbol de ese tipo para cada miembro de **P***. El problema con esta representación es inherente a la naturaleza de los miembros de **P**; cada uno de estos, es un segmento donde se registran las propiedades de interés para el individuo de **P**. Para relacionar a este individuo con varios otros en **Q**, se requieren tantos punteros como sean los miembros de **Q** involucrados.

Los grafos siguientes (parte izquierda del gráfico), muestran en forma aislada, un sólo individuo **p** de **P**, y cuatro individuos de **Q** con los cuales se relaciona el anterior **q1, q2, q3** y **q4**. Esta figura, muestra arcos diferentes desde **p** a cada uno de los puntos.

Una solución al problema de los punteros múltiples es conectar aquellos individuos de **Q**, que se relacionan con el mismo individuo **p**. Se trata en este caso de una relación diferente y puede observarse con un tipo de línea distinto (de puntos en el ejemplo de la parte derecha del gráfico). Aquí se tiene un tipo de puntero desde **p** a **q4** indicado por un arco de línea. A continuación, **q4** apunta a **q3** el cual apunta a **q2** y este a **q1**.



Relación padre - hijo

Se ilustrará a hora una relación uno a muchos, utilizando la expresión “*es el padre de*”. Se podría usar la expresión “*padre o madre de*”, pero un niño tiene dos padres e interesa mantener esta relación como uno a varios.

Un padre puede tener varios hijos, lo cual puede ser definido como:

$$p \text{ } F \text{ } q1; \quad p \text{ } F \text{ } q2; \quad p \text{ } F \text{ } q3; \quad p \text{ } F \text{ } q4$$

Donde **F** representa “*es el padre de*”. Estos hijos, mantienen una cierta relación entre sí; todos tienen el mismo padre natural. Resulta claro que estos hijos son hermanos, lo cual se puede definir del siguiente modo:

$$q4 \text{ } S \text{ } q3; \quad q3 \text{ } S \text{ } q2; \quad q2 \text{ } S \text{ } q1$$

Donde **S** es la relación “*es hermano de*”. Por lo tanto, **S** está representado por una línea de trazos como la vista en el último digrafo.

El problema que se presenta ahora es cómo se seleccionará el orden en el que los individuos de **Q** que se deben conectar. Para resolver esto, se puede imponer una restricción adicional a la relación entre hermanos (se la llamará **Y** cuyo significado sea: “*es el hermano menor de*”); por lo tanto, se tendrá:

$$q4 \text{ } Y \text{ } q3; \quad q3 \text{ } Y \text{ } q2; \quad q2 \text{ } Y \text{ } q1$$

Ahora, las líneas de trazos del último digrafo, pueden ser definidas como la relación de hermanos menores: **q4** es el menor de todos; **q3** es el siguiente en orden de minoridad y apunta a **q2**; y así sucesivamente. Dado que **q1** es el hermano de mayor edad, no tiene hermanos mayores. El puntero de **q1** contiene una señal terminal.

Las líneas de trazos del digrafo, pueden ser implementadas mediante una serie de punteros que conecten los miembros de **Q**. En este ejemplo, todos los hijos son de la misma población; el padre **p** está en otra población. Este apuntará a un arco de “color” distinto a su hijo menor **q4**.

RELACIONES ENTRE POBLACIONES

Problema

Una base de datos representa varias poblaciones, ya sea que ella sea utilizada por un sólo programa de aplicación o para una familia de aplicaciones. Una característica importante de la base de datos es que puede representar una forma de relación entre dos individuos cualesquiera. Ya se ha visto cuantas formas de relación pueden plantearse entre individuos de la misma población; también es necesario representar formas de relación que puedan existir entre individuos de distintas poblaciones.

Relaciones uno a uno entre Poblaciones

Se considerará primero el caso de dos poblaciones para las cuales existe una relación uno a uno entre individuos de ambas. Para dar forma completa a esta idea, considérese un depósito que contiene productos y su distribución física dentro del almacén; el cajón o estante: éste contiene el número de mercaderías que se tiene disponible. Los componentes de esta forma de relación: sujeto, objeto y relación son:

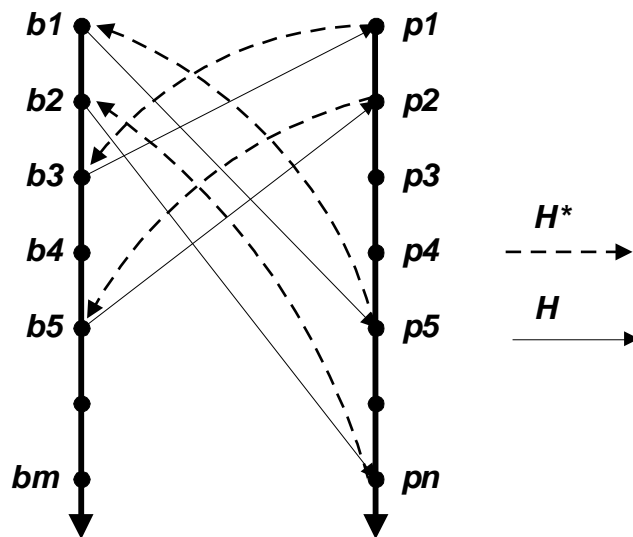
- **b** es la población de cajones o estantes
- **p** es la población de productos
- **H** es la relación “contiene”

H es la relación entre un cajón y los productos que contiene. Existe una relación inversa entre un producto y un cajón **H***. Se dirá en este caso que un producto, está almacenado o contenido en un cajón y, por lo tanto:

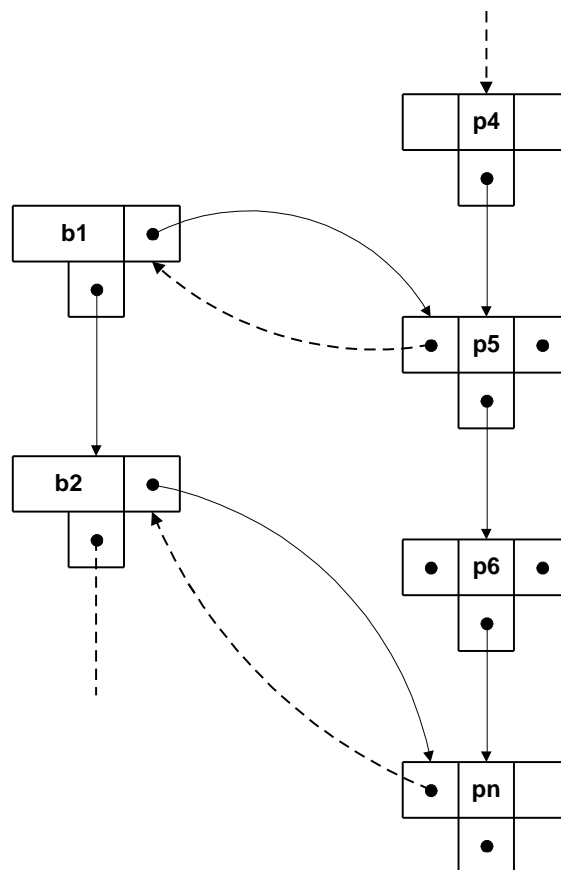
$$\mathbf{b\ H\ p; \quad p\ H^*\ b}$$

La figura siguiente ilustra esta situación. A la izquierda, se puede ver una línea gruesa dirigida con vértices que representan a los cajones: una población. A la derecha, se tiene otra línea con vértices que representan los productos, la segunda población. Entre ambas líneas verticales, existen arcos diagonales con líneas finas que representan la relación **H**. Por lo tanto, el primer cajón, **b1**, contiene los productos designados como **p5** y una línea fina de trazos apunta desde **p5** a **b1** para señalar que estos productos, están contenidos en dicho cajón y se tiene:

$$\mathbf{b1\ H\ p5; \quad p5\ H^*\ b1}$$



Los segmentos que representan los productos y aquellos que representan los estantes, están conectados entre sí dentro de su población por un conjunto de punteros, como lo muestra la siguiente figura:



El puntero de la parte inferior de cada segmento, lo conecta con el segmento sucesor, y en consecuencia provee un ordenamiento dentro de la población. Los “colores” y los tipos de punteros se corresponden: las líneas gruesas relacionan estantes, así como lo hacen los punteros en la parte inferior del segmento.

Puede observarse una relación entre un segmento correspondiente a un estante y un segmento correspondiente a un producto, por medio de un puntero ubicado a la derecha del individuo estante, el cual representa la relación ***H*** - líneas finas y punteros apuntando a la derecha se corresponden -. De la misma manera, se relaciona un segmento de producto con un segmento de estante por medio de un puntero a la izquierda de aquel, representando la relación ***H**** - donde la línea de trazos y los segmentos de la izquierda se corresponden -.

Relaciones de uno a muchos entre Poblaciones

Considérese nuevamente dos poblaciones; en este caso, un individuo de la primera, está relacionado con varios individuos de la segunda. Se tomará nuevamente el ejemplo del depósito que contiene productos y su distribución física dentro del estante. Se le agregará ahora, que *cada producto, está asociado con solamente un vendedor*, pero que *un vendedor* puede comercializar **varios** productos distintos. Para la relación “*provee*” se tendrá:

- Un vendedor, *v*, suministra varios productos.
- Un producto, *p*, puede ser vendido por sólo un vendedor.
- Sea *S* la relación “*provee*”.

Esta relación de uno a varios puede ser simbolizada como:

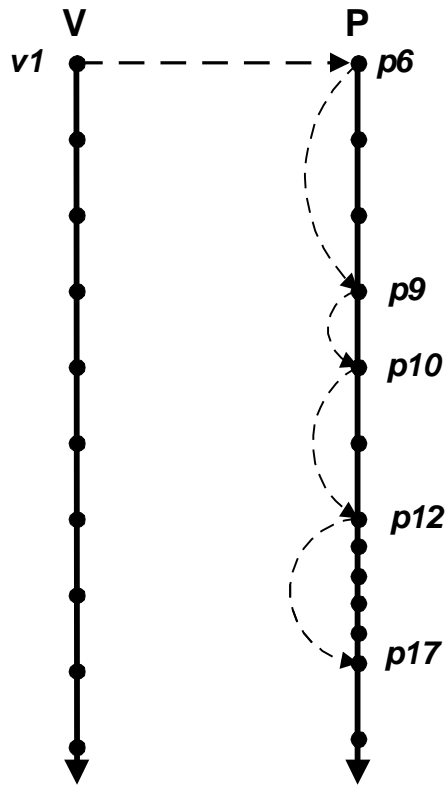
$$v \ S \ p; \quad v' \ \bar{S} \ p$$

Lo cual significa que un vendedor, provee un producto, que a su vez no es provisto por ningún otro vendedor.

Dado un vendedor, existe un número de productos que él mismo provee. Ese conjunto de productos vendidos por el vendedor **v1**, se llamará **p1**, y la numeración de este conjunto será:

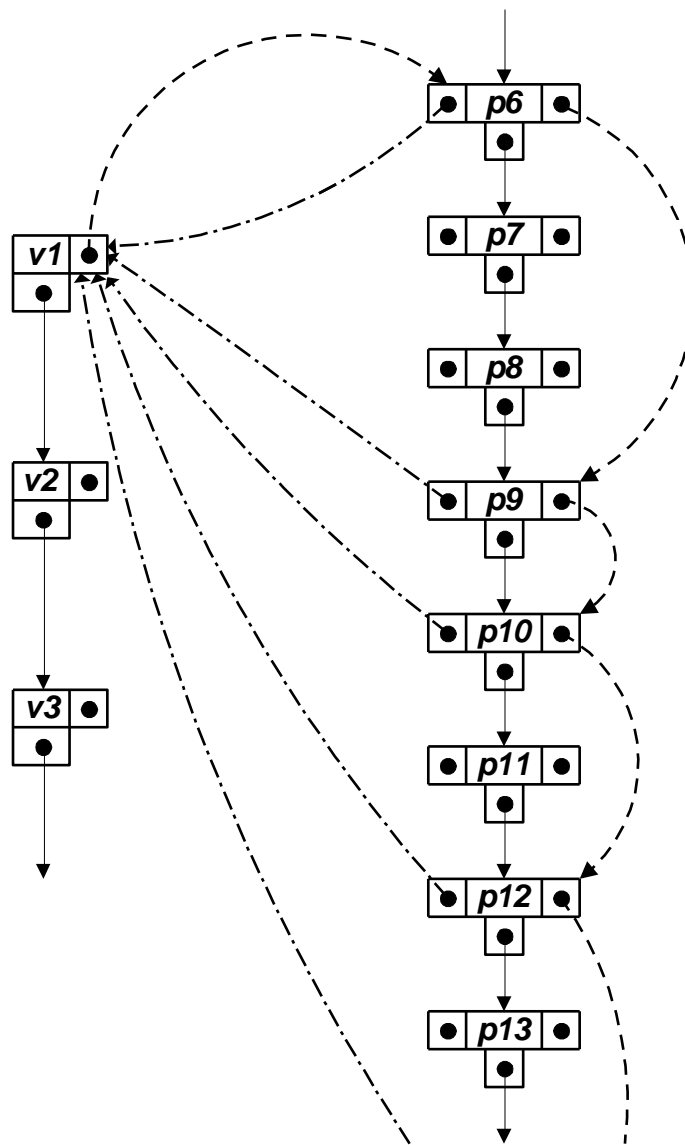
$$\mathbf{p1} = \{ p6, p9, p10, p12, p17 \}$$

La siguiente figura, ilustra esto gráficamente.



La población de vendedores aparece a la izquierda. Dicha población está **organizada** por la relación que se observa como línea gruesa (vertical). La población de productos está a la derecha. La misma tiene también una organización impuesta a sus datos, que puede ser distinta a la disposición física dentro del archivo. De cualquier manera, la línea gruesa también indica un tipo de organización.

Se relacionará ahora un vendedor con los productos que provee. Se elegirá el vendedor **v1** el cual vende los productos llamados **p6, p9, p10, p12** y **p17**. El arco que va de **v1** a **p6** es uno de los productos provistos por el vendedor **v1**. Otros productos provistos por **v1**, pueden ser apuntados desde éste, por ello requeriría un número variable de punteros asociados con el segmento **v1**. Por lo tanto, una alternativa interesante al número de variable de punteros sería asociar entre sí a los individuos de la población objeto. La figura, usa “color” para expresar gráficamente esta idea. El arco desde **v1** a **p6** está indicado en trazos gruesos; el mismo relaciona poblaciones. Los arcos entre **p6** y **p9**, **p9** y **p10** y así siguiendo, están indicando en trazos más finos para indicar esta distinta asociación. Como idea previa, para llevar a cabo esto dentro de la base de datos, véase la siguiente figura:



Aquí se tiene un número de productos asociados entre sí en la población por medio del puntero en la parte inferior de cada segmento, formando de este modo la línea vertical de la figura. Existe otra forma de encadenamiento de segmentos, que asocia los productos provistos por el mismo vendedor. Puede apreciarse en la figura que como mínimo los productos *p6*, *p9*, *p10* y *p12* son provistos por el mismo vendedor utilizando este puntero.

Dado un producto, hay solamente un vendedor que lo provee. La relación inversa a *S*, es *S**, que significa, “*es provisto por*”. Otro puntero en cada segmento de producto relaciona ese producto con su vendedor en la población de vendedores como se vio en la figura. Ahora, dado un producto, se puede encontrar su vendedor, tarea frecuente en este tipo de aplicaciones.

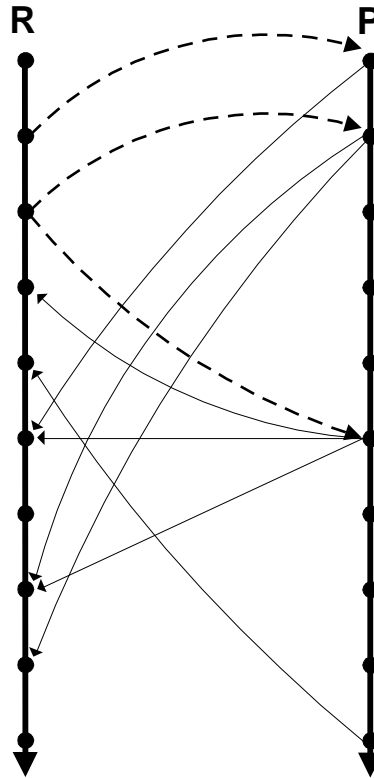
Relación muchos a muchos entre poblaciones

Resulta mucho más difícil representar relaciones muchos a muchos. Considerando nuevamente el ejemplo anterior pero después que la compra de productos ha sido ordenada mediante

la emisión de una orden de compra o de un requerimiento que se indicará como r . Un sólo requerimiento puede ordenar una cierta cantidad de productos, pero varios requerimientos, pueden haber sido emitidos para referirse al mismo producto. Por lo tanto, un producto puede aparecer en varios pedidos, y varios pedidos pueden referirse al mismo producto; para la relación orden de compra, O , se tiene:

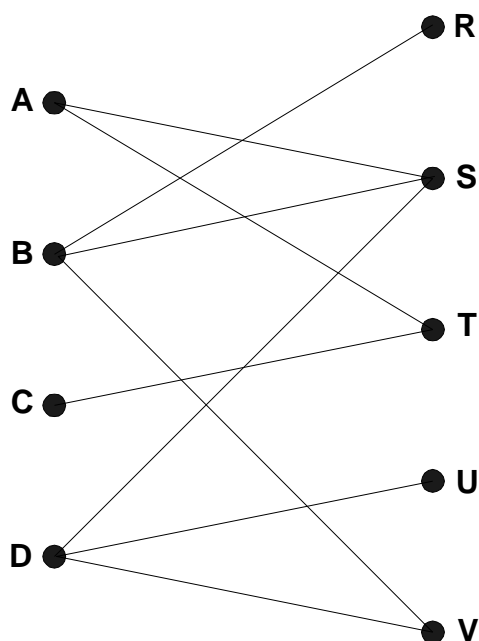
$$r \ O \ p; \ r \ O \ p'; \ r' \ O \ p$$

En la siguiente figura, puede observarse un grafo que relaciona las dos poblaciones



Funcionamiento de Conexión

Para describir el funcionamiento, se tomará un ejemplo abstracto basado en el gráfico de la siguiente figura:

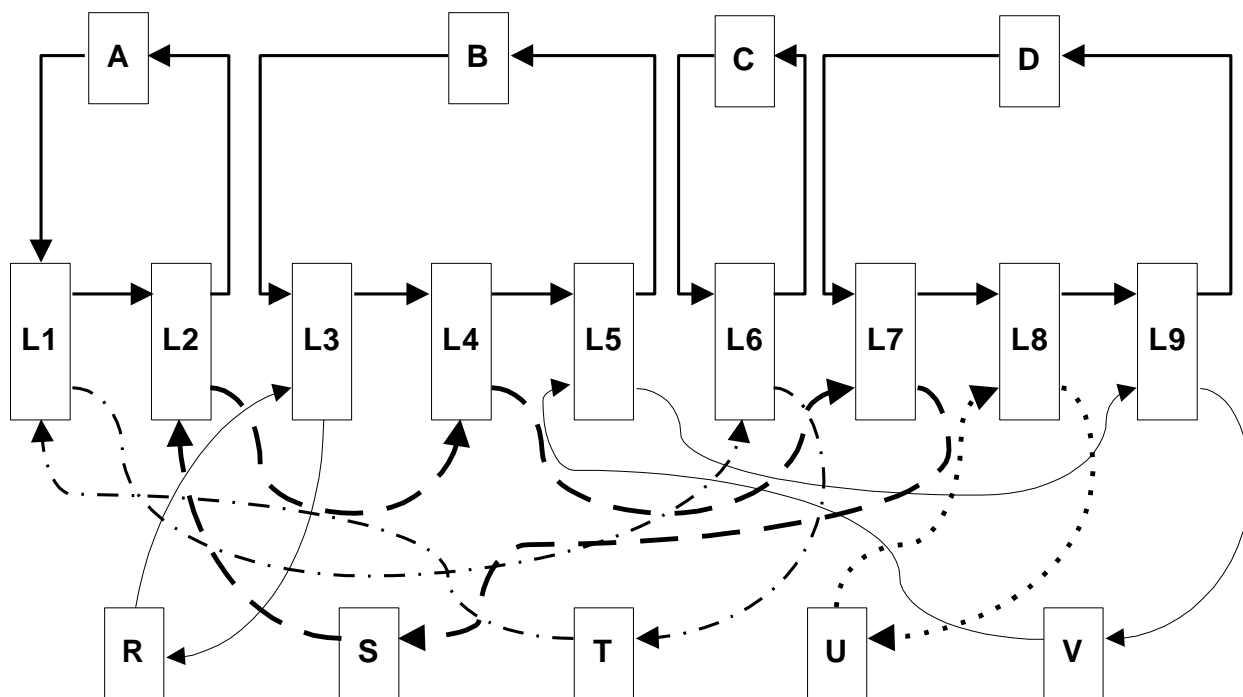


Los cuatro objetos representados por los vértices **A**, **B**, **C**, **D** sobre la izquierda, se relacionan con los cinco objetos **R**, **S**, **T**, **U** y **V** de la derecha. Las aristas entre los nodos indican cuáles de ellos están relacionados. Por lo tanto, la arista que va de **B** a **R** muestra que **B** y **R** están relacionados; dado que no hay arista entre **B** y **U**, estos dos no tienen relación alguna.

Se dirá que son sujetos las entidades de la izquierda, **IZQ** y objetos las entidades de la derecha, **DER**. Un tercer tipo de entidad sintética o artificial se llama **CONEXIÓN** **que es la que permite resolver la relación de muchos a muchos**. Cada sujeto de la forma de relación, tiene una conexión; lo mismo sucede con cada objeto. Se trata de una misma relación que cuando el sujeto y el objeto están relacionados entre sí. Se tiene entonces:

- **RI relaciona IZQ con CONEXIÓN.**
- **El conjunto RD relaciona DER con CONEXIÓN.**
- **Hay tantas CONEXIONES como formas de relación.**

Este último párrafo, da la guía para armar la ocurrencia de la figura que se representa en el siguiente gráfico:



Debe crearse la red de un modo sistemático (no es la única forma). Arrancar con el primer **IZQ**, **A**, y usar una conexión para cada **DER** con la cual se relaciona **A**. Las conexiones, **L1** y **L2**, unidas a **A**, representan las dos **DER**, **S** y **T** con las cuales se relaciona **A**. De un modo similar **L3**, **L4** y **L5** pertenecen a **B**, y relacionan **B** con las tres **DER**: **R**, **S** y **V**. **C** posee solamente **L6** y **D** tiene **L7**, **L8** y **L9**. Cada dueño, debe estar al comienzo del bucle que contiene sus conexiones.

Luego de registrar las conexiones, se conectarán las mismas con las **DER**. Se debe hallar una conexión para cada **IZQ**, con la que esta **DER** se relaciona y unir estas conexiones entre sí en un bucle. Esto puede demostrarse mejor en el ejemplo. Se arranca con **R**. Esta se relaciona solamente con **B**. Se conecta **R** a **L3**, la cual establece esta forma de relación. **S** se relaciona con tres **IZQ**, **A**, **B** y **D**. Se debe encontrar tres conexiones, una en cada uno de los conjuntos que pertenecen a **A**, **B** y **D**, y conectar las mismas entre sí en un bucle con **S**. Se tendrá entonces:

- **S** apunta a **L2** que pertenece a **A**;
- **L2** a su vez, apunta a **L4** que pertenece a **B**;
- **L4** apunta a **L7** que pertenece a **D**;
- **L7** apunta nuevamente a **S** para establecer la pertenencia de **DER**.

Es importante tener en cuenta que:

- Cada **IZQ** tiene un puntero que se dirige a una **CONEXIÓN** la cual es el comienzo de una cadena para todos los miembros que le pertenecen.
- Cada **DER** tiene un puntero que se dirige a una **CONEXIÓN** que es el comienzo de una cadena para los miembros que pertenecen a dicha **DER**.
- Cada **CONEXIÓN** tiene dos punteros de distinto tipo; el puntero superior, corresponde a su dueño en **RI**; el puntero inferior a su dueño **RD**.

Orden

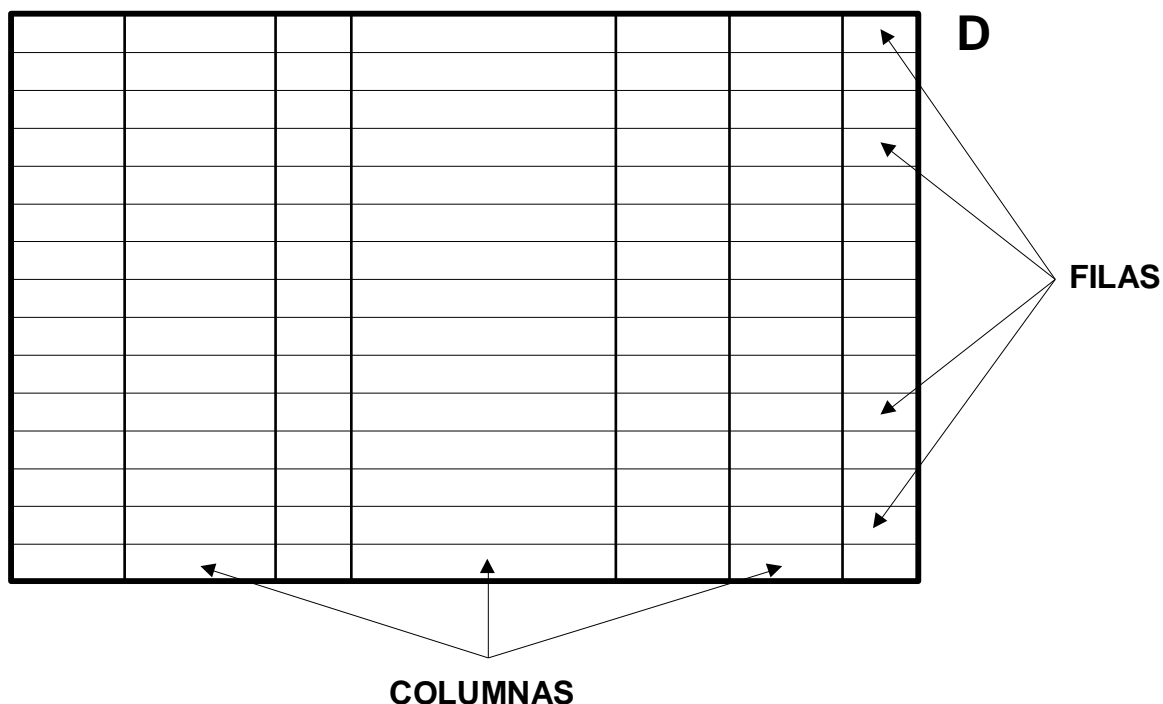
Es posible imponer un orden sobre las **CONEXIONES** que correspondan tanto a la clave **IZQ** o a la **DER**, o incluso a ambas. Por ejemplo, para el objeto **S** en la figura, las conexiones son ordenadas por su clave **IZQ**. Por lo tanto, **L2** se encuentra en primer lugar, y corresponde a **A**; **L4** es la próxima y corresponde a **B**; **L7** es la última y corresponde a **D**; el orden de claves puede existir simultáneamente en el otro bucle. Tomando **B**: posee conexiones que la relacionan con **R**, **S** y **V**; estas conexiones son **L3**, **L4** y **L5**. Su orden corresponde al orden de la clave del objeto.

LA BASE DE DATOS RELACIONAL

Cualquiera sea el tipo de arquitectura utilizada, siempre será necesario establecer conjuntos de relaciones entre dos o más elementos conceptuales. Se observa claramente que los elementos conceptuales si bien son abstractos, representan “algo”, mientras que las relaciones no resultan tan tangibles ya que no pueden ser medidas o clasificadas con claridad como los atributos de un individuo (de una clase o una entidad). Una relación no existe como algo separado o tangible. Los datos son abstractos, pero más aún lo son las relaciones. Dentro de la computadora, los datos pueden existir como señales magnéticas o eléctricas. Pero cuando los datos se convierten en parte de una base de datos, ésta incluirá las relaciones que existen entre las entidades de datos, de modo tal que las relaciones también se deben convertir en algo concreto; vale decir que se da a la relación un nivel equivalente a los datos a los cuales se aplica. La relación se formula del mismo modo que los datos. Por lo tanto, esta relación se convierte en algo tan concreto como aquellos datos. La forma de trabajar con ambos, datos y relación, es la tabla o matriz que se desarrollará a continuación.

LA MATRIZ DE DATOS

La base de datos relacional fue creada por personas orientadas matemáticamente. Ellos deseaban obtener una visión uniforme de los datos. Todos los datos acerca de una población se ingresan en una sola tabla. Esta tabla se denomina también *matriz*. Como ejemplo de este tipo de tablas, se presenta el de la siguiente figura:



La citada tabla es bidimensional; consiste en filas y columnas:

- Una *fila* comprende todos los datos que están en la misma línea horizontal
- Una *columna* comprende todos los datos dispuestos sobre la misma línea vertical

Un dato simple, es denominado *celda*. Una fila consiste en una cierta cantidad de celdas horizontales. Una columna es el conjunto de celdas verticales. Se puede identificar una celda, dando como dirección la fila y la columna en que se halla ubicada.

Se utilizarán letras mayúsculas en negrita como símbolo de una tabla. Se denomina **D** a la tabla o matriz de la figura anterior.

La matriz **D** es análoga a un archivo. Cada fila corresponde a un registro, consistente en un grupo de campos. Por lo tanto, un campo y una celda son conceptos comparables. La posición de una celda dentro de una fila, es importante. Se identificará qué representa dicha celda por medio de su posición dentro de la fila. Todas las filas consisten en conjuntos ordenados de celdas. Por otro lado, la posición de una fila dentro de una matriz, no es importante; dos matrices se denominan equivalentes, siempre que consistan en las mismas filas, independientemente del orden en el cual éstas aparezcan en cada una de las matrices.

Relación uno a uno

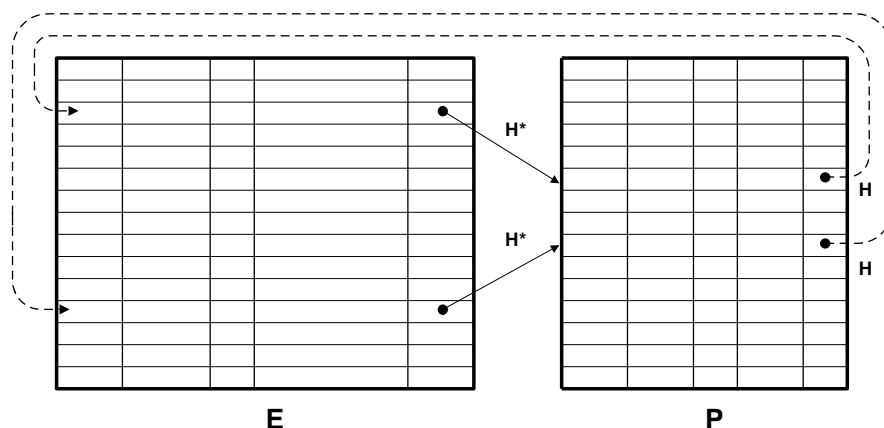
Se considerarán ahora dos poblaciones. Cada una de ellas, está representada por su propia matriz. Para una forma de relación uno a uno, se trata de un problema simple, ya que para cada fila de una matriz, existirá una fila correspondiente en la otra matriz, y todo lo que se tiene que hacer, es establecer punteros entre las filas correspondientes.

Ejemplo:

Se considerarán las poblaciones de *estudiante* y *persona*. La población *estudiante* es un subconjunto de *persona*. El vínculo que mantienen es a través de una relación de funcionalidad 1-1. En la siguiente figura, puede verse que la población de *personas*, se halla representada por la matriz **P** de la derecha. Los estudiantes están representados por la matriz **E** de la izquierda. Para relacionar un estudiante con una persona, se usará la forma de relación **H***. Por lo tanto, **H*** relaciona una fila de **E** con una fila diferente de **P**. Esta asociación resulta en un puntero desde una fila de una población hacia otra fila en la otra población.

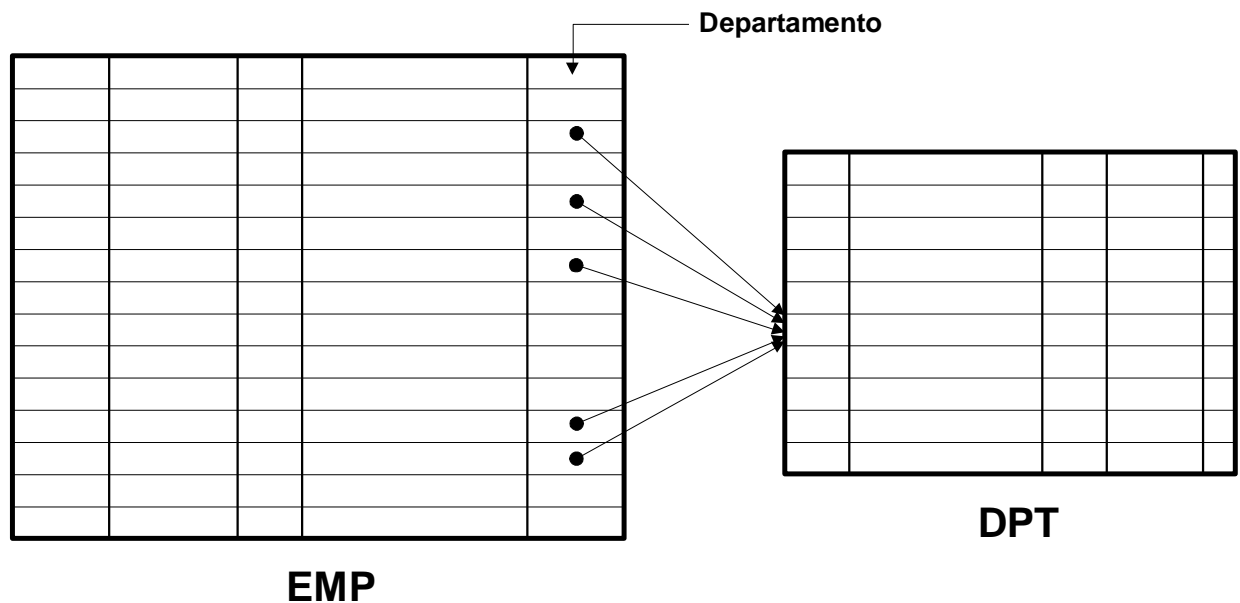
Hay una diferencia definida en los medios por los cuales se asocia una fila con otra; se trata de algo simbólico más que físico. Cada *estudiante* (fila), contiene una celda destinada a almacenar la *persona* de su identidad. Esta representación de la persona, es por ejemplo el documento (clave). Para encontrar esta fila, se debe buscar en la matriz de personas, entrando por la clave que identifica un individuo en particular.

Puede ser necesario proveer un efecto recíproco, vale decir, dado una persona, encontrar los datos del estudiante (si es que la persona lo es). Esto se lleva a cabo de la misma manera. Se necesita ahora que cada fila de **P** contenga un número de estudiante (clave) que podría ser el número de libreta universitaria. Este número de libreta universitaria permite encontrar la fila que describe al estudiante que es esa persona. Una instancia de dicha relación **H** se ilustra mediante la línea de trazos de la figura.



Muchos a uno

La relación muchos a uno, puede tratarse de la misma manera. Considérese un empleado asociado a un departamento en particular. Es posible construir dos matrices, una para la población *empleado* y otra para la población *departamento*. En el siguiente ejemplo, se llamarán **EMP** y **DPT** respectivamente. Esto puede observarse en la figura siguiente:



Varios empleados se encuentran asociados exactamente con un departamento; por lo tanto, es posible colocar en cada fila de empleados, un número de departamento. Varias filas tendrán el mismo número de departamento como se observa en el gráfico. Todos estos empleados trabajan o están asociados al mismo departamento. Esta disposición tiene la ventaja de que sólo existe un número de departamento y de que la información del departamento no se repite para cada empleado. Solamente como comentario se acota que esta estructura de tablas no permitiría llevar un registro histórico de empleados, vale decir, la lista de todos los departamentos por los que pasa un empleado a lo largo de su carrera administrativa, pues, para ello, debería haber tantas columnas en la tabla **EMP** como departamentos distintos hayan sido asignado cada uno de los empleados. Puede presentarse el caso de que un empleado haya pasado tan sólo por un departamento, y otros por **n** departamentos distintos; consecuentemente el escenario mencionado constituye otro tipo de funcionalidad en la relación.

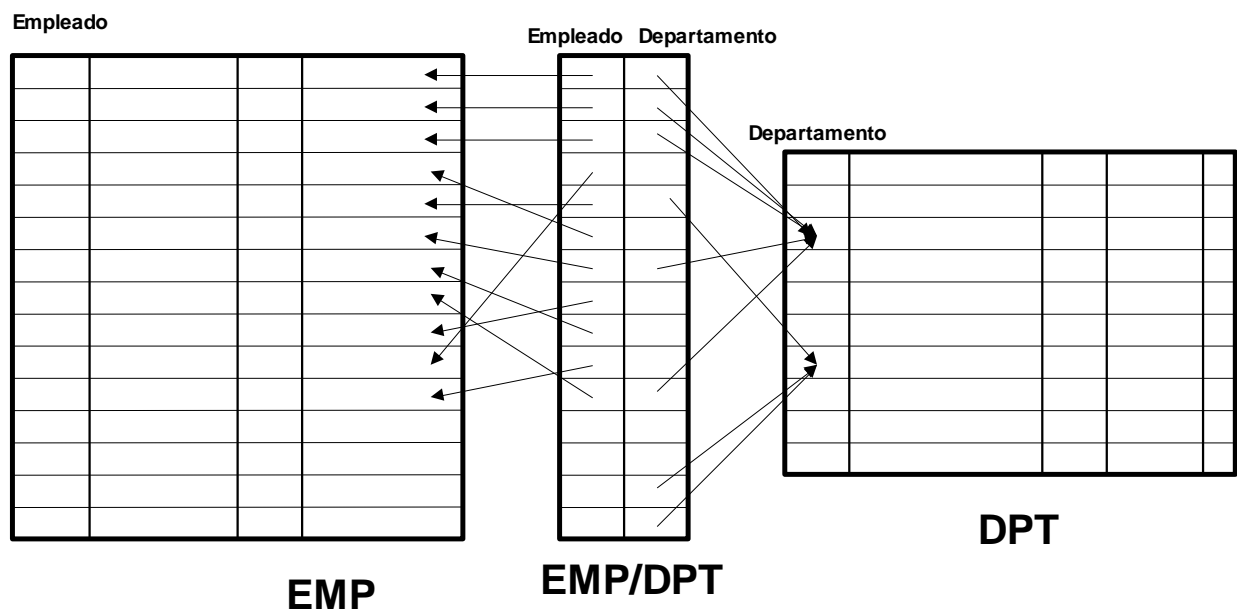
Uno a Muchos

Ya se ha visto cómo encontrar la descripción del departamento siempre que se haya localizado a un empleado. Para invertir esta situación, dado un departamento, se puede desear encontrar todos los empleados que trabajan en el mismo. No es posible llevar a cabo esto con dos tablas si se desea mantener un número fijo de filas y columnas en cada una de ellas. Se hace la salvedad que, a través de una aplicación, puede recorrerse la tabla de empleados y levantar aquellas filas en las que la celda que indica el departamento en el que trabaja, coincida con el seleccionado. Existe otra alternativa: **crear otra tabla** cuyo sólo propósito sea **asociar** empleados con departamentos, y se la denominará **Matriz Relacional**.

El uso de una matriz adicional para indicar la relación que existe entre dos tipos de individuos, se ilustra en la siguiente figura. Nuevamente se tienen las dos matrices, **EMP** y **DPT** para empleados y departamentos. La matriz agregada **EMP/DPT** contiene solamente dos columnas, una para el número clave del empleado y otra para el número clave del departamento. Cada

número de departamento puede aparecer en varias filas diferentes, mientras que cada número de empleado es único.

Para encontrar en qué departamento trabaja un empleado, se necesitará consultar solamente **EMP/DPT**. Inversamente, si se desea encontrar los empleados asignados a un departamento, también se puede consultar esta tabla. Para obtener información combinada acerca de un empleado, se debe entrar en **EMP/DPT** para localizar la descripción del departamento y su propia descripción identificando las filas correspondientes en **EMP** y **DPT**. Se pueden usar estas tablas de un modo similar para clarificar la información acerca de todos los miembros del departamento o bien para hacer más eficientes las búsquedas en términos de tiempo.



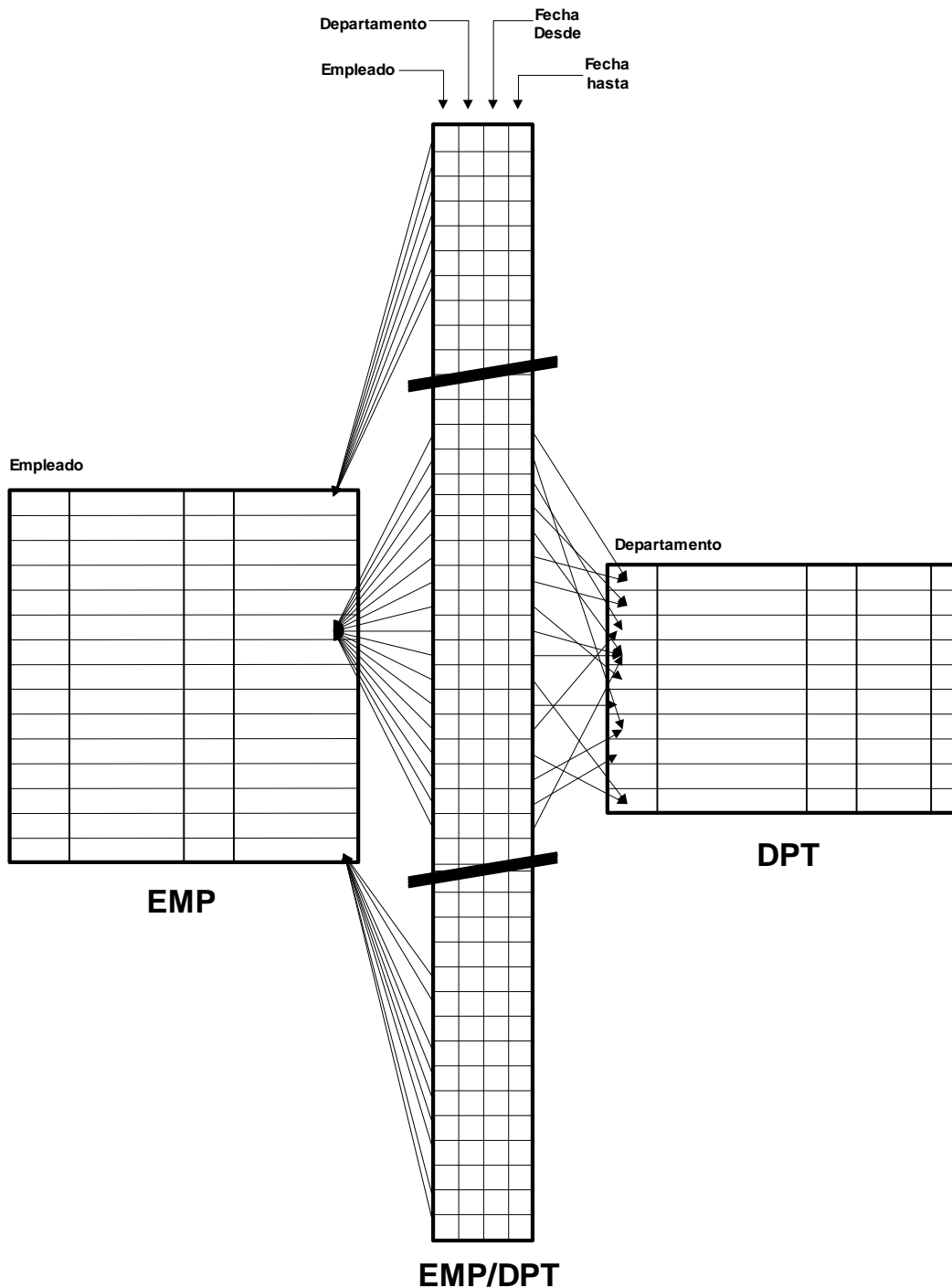
Nótese que existen tantas filas de la matriz relacional, como filas hay en la tabla de empleados. Esto resulta lógico, pues todos los empleados deben estar relacionados con un departamento.

Muchos a Muchos

Resulta relativamente fácil extender este concepto, a partir de la relación uno a muchos a la relación muchos a muchos. Una tabla de relación como esta, sigue siendo necesaria para relacionar dos tablas de datos. Considérese ahora, que se desea tener en cuenta la historia del empleado en lo que respecta a las distintas reparticiones por las que fue pasando a lo largo de su carrera. Ello conduce a una relación de muchos a muchos, puesto que un empleado por ejemplo, puede haber recorrido varios departamentos y aun puede repetirse su situación pero para dos fechas diferentes, vale decir que la fecha formará parte del identificador.

En el gráfico de la siguiente página, puede observarse la manera de resolver tal caso. Nótese que para una fila de la tabla **EMP** puede haber varias en la tabla de relación **EMP/DPT**, y que ocurre la misma situación haciendo referencia a la tabla **DPT**.

A manera de acotación, si se desea obtener la lista de los empleados que pertenecen a un departamento en la actualidad, puede recurrirse a la tabla de relación y obtener el subconjunto de las filas en que figure el departamento y que además, la celda que representa la *fecha hasta* (fecha de baja del empleado en ese departamento) no contenga ningún valor. Asimismo, si la lista deseada es la historia de un empleado en particular, puede recurrirse a la tabla, y consultar mediante el número del empleado, el subconjunto de filas en las que coincida el número del empleado buscado sin importar ninguna otra columna. Obviamente, en ambos casos, tanto para determinar los datos del empleado y los del departamento, debe recurrirse a las tablas **EMP** y **DPT**.



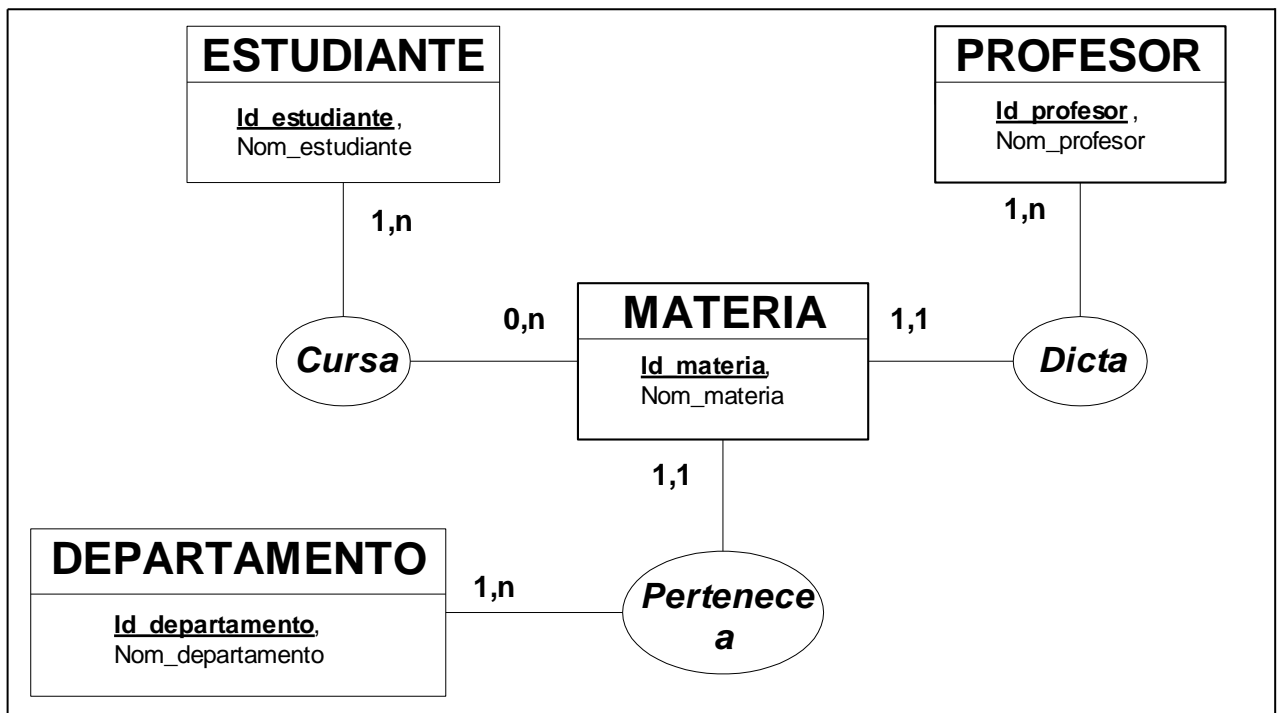
La Base de Datos “Alumnado”

Con el propósito de ejemplificar una base de datos simple, se considerará un sistema de alumnado de cierta facultad. Se definirá como objetivo principal, el dictado de clases a los estudiantes. Las clases son por lo tanto el elemento de mayor importancia. Se hará una serie de simplificaciones iniciales dadas por las siguientes reglas de gestión:

- Un PROFESOR dicta al menos una MATERIA
- Una MATERIA es dictada por uno y solamente un PROFESOR
- Un ALUMNO puede estar inscripto y cursar varias MATERIAS
- Una MATERIA depende de uno y solamente un DEPARTAMENTO

Modelo conceptual de datos de “Alumnado”

De acuerdo a las reglas de gestión, MCD correspondiente es:



El Registro Compuesto

Una forma directa de construir una base de datos, es generar un registro compuesto. El registro compuesto, tiene toda la información con respecto a cada conjunto existente de formas de relación que afecten a todas las variables - estudiantes, profesores, materias y departamentos-. La cuestión es cómo hacer esto de modo de obtener un agrupamiento significativo.

Para un registro compuesto, se necesita una forma de relación compuesta; para el ejemplo, esta asociación existe entre un estudiante y un profesor dentro de una única materia. Una asociación múltiple de este tipo, es un "individuo" en el contexto de un registro compuesto. Debe tenerse en cuenta que no se trata simplemente de alguna asociación entre un estudiante y un profesor; un

estudiante puede cursar dos materias dictadas por el mismo profesor, tratándose en este caso de dos asociaciones distintas. Cada tupla de esta forma de relación, es un individuo que tiene un registro para describirlo.

La figura siguiente, permite observar porciones de un registro típico de esta población. Debe recordarse que se desea tener un registro para cada asociación, que describa los individuos que toman parte en dicha asociación; esto vale para sus muchos campos. Tal como se plantea el problema, no parece que se tenga una clave única para este registro: ninguno de los participantes - estudiantes, materias o profesor -, provee unicidad para esta asociación. Existen dos formas para resolver el problema:

- Utilizar un registro compuesto consistente en una identificación concatenada con elementos significativos de materia y estudiante
- Asignar al estudiante un número único distinto para cada una de las materias que cursa. En la figura, esto aparece como una matrícula ID estudiante-materia, denotando la clave compuesta

Por otra parte, se necesita también una identificación única para cada materia, ID materia, que es el atributo que la identifica. Éste será distinto para cada descripción de materia. La combinación a adoptar en referencia a la relación con los estudiantes que la cursan, podría consistir en la matrícula Id_ materia, más el número ordinal del estudiante según su inscripción en tal materia. Por otra parte, y en virtud de ser estudiante, un elemento fundamental y obviamente una entidad, deberá existir alguna manera de identificarlo unívocamente, y no a partir del número ordinal de inscripción en una materia, que dependiendo de ésta última, cambiará. El identificativo adoptado para el estudiante será Id_estudiante. Algo similar, ocurrirá con los profesores y con los departamentos.

ID Estudiante Materia	ID Materia	Nombre Materia	ID Departamento	Nombre Departamento	ID Profesor	Nombre Profesor	ID Estudiante	Nombre Estudiante

Para identificar al estudiante, se puede utilizar una codificación particular de la Facultad. Puede usarse el tipo y número de documento, número de libreta universitaria, etc. Para identificar al profesor, también debe recurrirse a alguna forma de clave.

Puede observarse que este registro compuesto, tiene toda la información que se puede necesitar para muchas aplicaciones. Ese precisamente es el problema, hay demasiada información repetitiva. La descripción de la materia, se encuentra repetida para cada estudiante. Lo mismo ocurre con la descripción del profesor y con la de los departamentos. Esto no resulta eficiente en términos de espacio.

Descomposición

Partiendo de un registro compuesto, se puede ahorrar mucho espacio conservando los datos sobre profesores, estudiante, materias y departamentos en conjuntos y separados. Si se procede de este modo, es necesario encontrar una forma de recolectar los datos más tarde, para obtener nuevamente el registro compuesto o un registro menor. Por ejemplo, el encargado de confeccionar las listas de alumnos regulares o en condiciones de cursar, o bien de hacer las actas de examen, puede desear saber solamente cuántos estudiantes están inscriptos en una determinada materia, y qué profesor imparte la misma. Todo lo que necesita, son los nombres de los estudiantes y del profesor y/o los números, y no todo el conjunto de información que también estaría disponible. Análogamente, para efectuar una interrogación acerca de qué materias está cursando un estudiante, el registro compuesto no solamente es ineficiente con respecto al espacio de almacenamiento, sino también con respecto a una recuperación rápida.

La base de datos relacional de “*Alumnado*”

El elemento básico de una base de datos relacional (BDR) es denominado tupla, forma abreviada de n-tupla, y es definida como un conjunto ordenado de n valores. Existe una tupla para cada individuo de interés en cada población de aplicaciones.

El conjunto de tuplas que describe a la población se define como la relación. Si bien se ha utilizado el término "relación" para describir cómo dos individuos actúan o están situados uno con respecto al otro, se trata aquí de un contexto diferente; no debe surgir confusión entre estos dos usos del mismo término. La relación para la BDR es una colección de tuplas, y será representado por una matriz. Existe una colección particular correspondiente a la población de aplicación. Cada tupla, se identifica para correlacionarla con un individuo. El componente o los componentes necesarios en cada tupla para este propósito, constituyen la clave primaria. Si bien las tuplas tienen una clave, la colección de tuplas, no es necesariamente un conjunto ordenado. Dos relaciones pueden verse como idénticas si contienen las mismas tuplas, independientemente del orden en el cual éstas aparezcan.

Una relación lleva por sí misma a la normalización -la base para la descomposición- o sea a la conversión de una relación en distintas sub-relaciones, las cuales tomadas en conjunto, preservan el contenido de información de la relación original pero en un formato más eficiente. Una base de datos relacional simple, es por lo tanto, un agrupamiento de sub-relaciones, obtenido a partir de la relación original, y tal cual se requiere para una sola aplicación o conjunto de aplicaciones. Una base de datos relacional, es una colección de estas relaciones simples.

La tabla de relación

Una relación es presentada a menudo, para la comunicación entre usuarios, como una tabla. Una tabla de relación de este tipo para el ejemplo, se muestra en la figura siguiente.

Id Materia	Nombre Materia	Id Depto.	Nombre Depto.	Id Profesor	Nombre Profesor	Id Estudiante	Nombre Estudiante
10725	Ing. de Software	100	Sistemas	1	CERI, José	666	SCRUTTI, Claudio
						127	MORETTI, A. R.
					
						25466	ROSSI, Fabiana C.
10726	Bases de datos	100	Sistemas	2	CERI, José	1433	SANTO, Gabriel M.
						25466	ROSSI, Fabiana c.
					
						3155	DEYRO, B.M.W.

La tabla se descompone horizontalmente en filas y verticalmente en columnas. Cada fila contiene una tupla, la cual corresponde a un registro. Cada una de las columnas, representa un atributo. Cuando se observa una tupla correspondiente a un individuo, el contenido de cada componente representa un valor de atributo que califica a ese individuo.

La forma usual de utilizar la base de datos relacional, es en primer término, armar la matriz compuesta, una tabla que contiene toda la información requerida para la aplicación. Cada fila, contiene una tupla compuesta. Esta tabla única, es ineficiente y repetitiva, como se ha visto en el Ingeniería de Software I, por lo que se hace necesario normalizarla para simplificar y descomponer la tabla en varias tablas más simples. De acuerdo a lo especificado inicialmente, el identificador de cada ocurrencia de la relación establecida entre el estudiante y la materia que cursa estará dada según la siguiente tabla:

Id Materia Estudiante	Id Materia	Nombre Materia	Id Depto.	Nombre Depto.	Id Profesor	Nombre Profesor	Id Estudiante	Nombre Estudiante
10725-1	10725	Ing. de Software	100	Sistemas	1	CERI, José	666	SCRUTTI, Claudio
10725-2	10725	Ing. de Software	100	Sistemas	1	CERI, José	127	MORETTI, A. R.
.....	10725	Ing. de Software	100	Sistemas	1	CERI, José
10725-n	10725	Ing. de Software	100	Sistemas	1	CERI, José	25466	ROSSI, Fabiana C.
10726-1	10726	Bases de datos	100	Sistemas	2	CERI, José	1433	SANTO, Gabriel M.
10726-2	10726	Bases de datos	100	Sistemas	2	CERI, José	25466	ROSSI, Fabiana c.
.....	10726	Bases de datos	100	Sistemas	2	CERI, José
10726-n	10726	Bases de datos	100	Sistemas	2	CERI, José	3155	DEYRO, B.M.W.

Subtuplas

Para simplificar aún más el modelo, se omitirá por ahora la presencia de la entidad **Departamento**, considerándose solamente las restantes. Se definirá ahora una subtupla como uno o más componentes de una tupla compuesta, asociados entre sí de algún modo funcional. En la figura anterior se distinguen tres subtuplas separadas por líneas verticales gruesas. Cada subtupla corresponde a una de las entidades de la aplicación; existen subtuplas para materia, estudiantes y profesores.

Subsiste un campo que es único y que por lo tanto, sirve como clave. Se trata de la matrícula **ID Materia - Estudiante**, un número compuesto en la figura; la primera parte es el número que identifica unívocamente a cada materia; la segunda parte es el número ordinal del estudiante dentro de la clase. El segundo número puede ser asignado de acuerdo con el orden en el que se han registrado los estudiantes. Por lo tanto, la matrícula 10725-1, corresponde a la materia identificada como 10725, y al estudiante que se ha registrado primero en dicha materia.

En la figura, los datos acerca de la materia así como acerca del profesor, permanecen constantes en todas las filas que pertenecen a una misma materia. Existe una fila diferente que describe a cada estudiante. En una tabla es común dejar espacios vacíos en las filas correspondientes a sub-tuplas duplicadas; la primera fila contiene la subtupla a ser duplicada en las filas en blanco que siguen debajo, como se muestra en la primer figura. Éste se repite sistemáticamente en la última figura.

Una tabla con partes de sus filas en blanco, como la de la izquierda, obviamente es no normalizada. En las técnicas de las bases de datos relacionales, es un requerimiento que todas las tablas, y por lo tanto todas las matrices, tengan valores en cada componente de cada tupla.

No normalizada



1FN

La tabla no normalizada de la izquierda, ha sido convertida en primera forma normal (**1FN**) una vez que todos los campos en todas las filas contengan un valor. Se dice que la tabla MEP (Materia - Estudiante - Profesor) está en su primera forma normal, ya que cumple con este requerimiento. Un requerimiento adicional de la primera forma normal, es que toda fila, tenga una única clave identificadora. Para el ejemplo, se trata en este caso de la matrícula ID materia-estudiante. Es posible elegir una clave compuesta a partir de atributos existentes sin necesidad de una clave sintética e independiente. Para poner esto más claro, considérese que una única clave para cada fila, podría consistir en una combinación de número de matrícula y número de estudiante. Se ha utilizado un número de matrícula de cinco dígitos; si el número de estudiante es el número de documento, se trataría de ocho dígitos adicionales. Más aun, no se tendría información del "orden de llegada" con respecto al estudiante (si realmente es de interés).

Descomposición

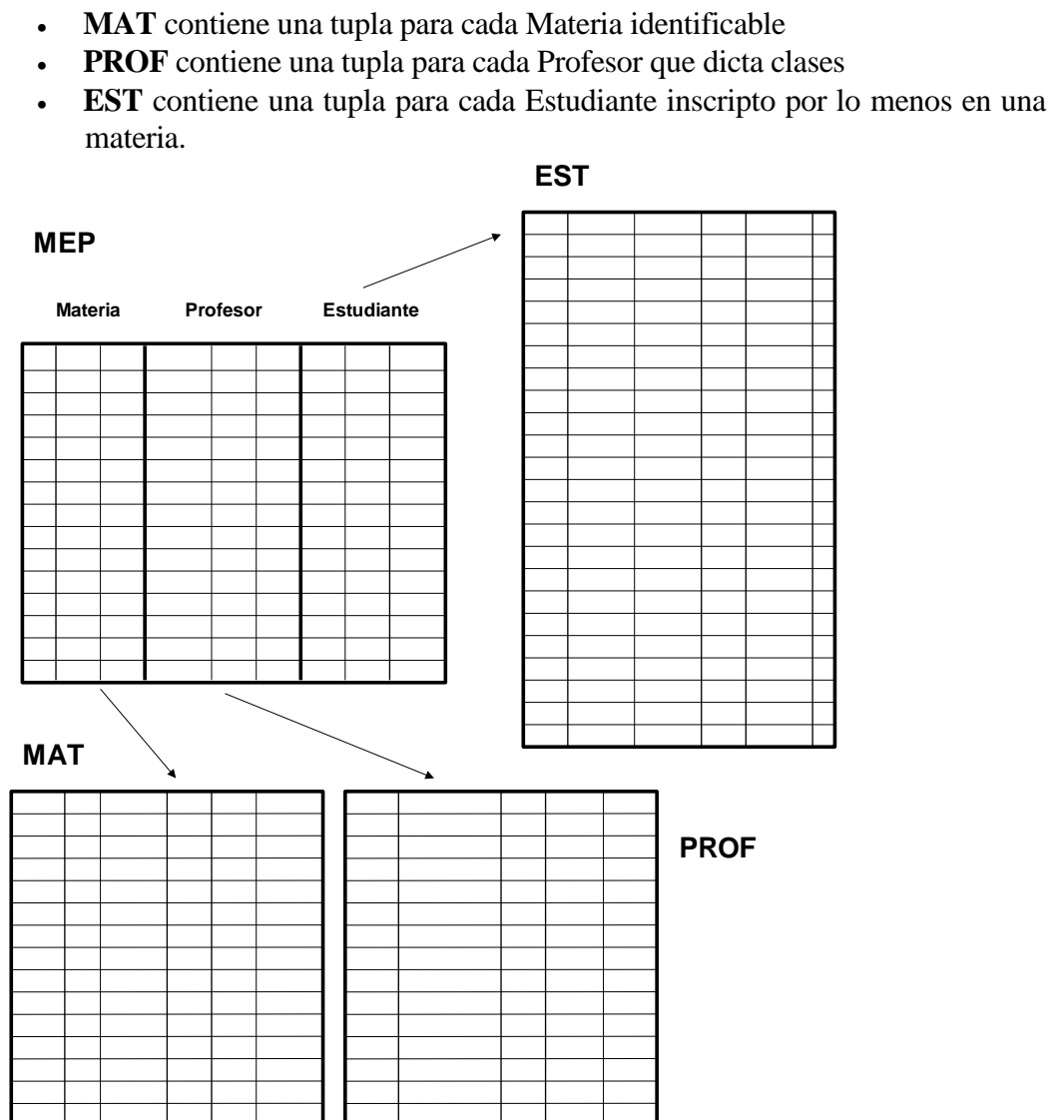
El primer paso en la descomposición, es identificar las subtuplas apropiadas. Dado que el ejemplo, incluye materias, profesores y estudiantes, deben construirse subtuplas adecuadas para cada una de estas entidades, como se ve en la siguiente figura. Se puede observar el conjunto origi-

nal en su primera forma normal como la tabla MEP. Las líneas horizontales gruesas, separan las tuplas de una clase con respecto a las tuplas de la clase siguiente. Las líneas verticales, definen las subtuplas. Las subtuplas han sido divididas a su vez en componentes, pero las columnas no tienen denominación alguna.

A partir de la tabla MEP se puede ahora obtener tres tablas más pequeñas cortando dichas tablas a lo largo de las líneas verticales gruesas. Antes de hacerlo, se debe asegurar que cada tabla tenga su columna clave, con un atributo que pueda identificar en forma unívoca a cada fila. Esto da lugar a tres tablas:

- La tabla de Materias **MAT**
- La tabla de Profesores **PROF**
- La tabla de Estudiantes **EST**

Como ya se ha explicado, cada tabla puede tener muchas filas duplicadas; éstas deben ser eliminadas. Cuando no existan duplicaciones, las tablas revisadas, aparecerán en la forma de figura siguiente, representando las nuevas relaciones:



Otra manera de describir la creación de la subtupla es la siguiente. Barriendo la tabla MEP, descomponer cada fila en subtuplas; distribuir las mismas en tres subtablas y eliminar cualquier subtupla para la cual exista un duplicado en la tabla.

Los requerimientos para la subtabla prevalecen respecto de la tabla original:

- **Cada nueva tabla está en la primera forma normal**
- **Cada tupla tiene una clave**

La eficiencia ganada en espacio da como resultado menores requerimientos de éste, y menor número de columnas para cada sub-relación, así que:

- Cada sub-relación tiene menor número de columnas
- Cada relación tiene por lo general menos filas, dado que el número de entidades para cada sub-relación es a menudo menor que el de la relación original

La tabla de estudiantes **EST** es la mayor y consiste en tantas filas como estudiantes haya. Esto contrasta con MEP que contiene una fila por estudiante y por materia que éste curse, lo cual es necesariamente mucho mayor.

Relacionando las subtablas

La relación original MEP relacionaba las tres entidades: Materia, Estudiante y Profesor. **MAT**, **EST** y **PROF** proveen toda la información previa excepto por el hecho de que no indican qué estudiantes cursan cada materia, ni qué profesor enseña en cada materia por lo que obviamente, falta algo más que permita restaurar la información original proveniente del registro compuesto.

La siguiente figura representa esquemáticamente cómo la relación adicional **MAT/EST** relaciona estudiante y materias. Las entradas en **MAT/EST** están dispuestas por materias y dentro de éstas por estudiantes. Nuevamente valen los requerimientos para una relación:

- **Está en la primera forma normal**
- **Cada fila tiene un único identificador**

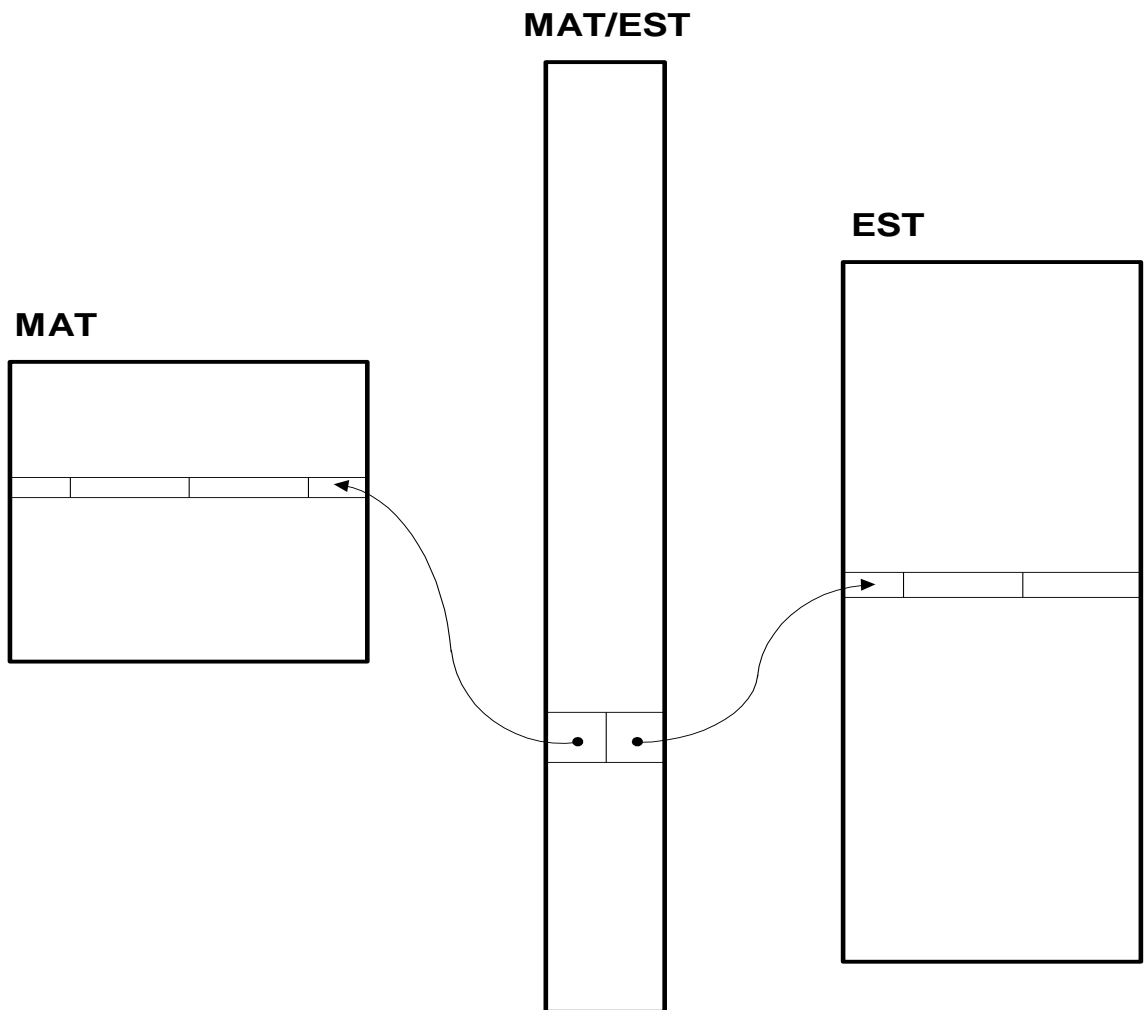
Se tiene en este caso una clave compuesta por los identificadores de materia y de estudiante. Cada fila, necesita contener solamente eso, dado que su propósito es relacionar las otras dos sub-relaciones. Es sin embargo un detalle importante, que una fila pueda contener mayor información, tal como el registro de asistencia y las notas en la materia. Debe enfatizarse que se trata sin embargo de información adicional y no un requerimiento de esta tabla de relación.

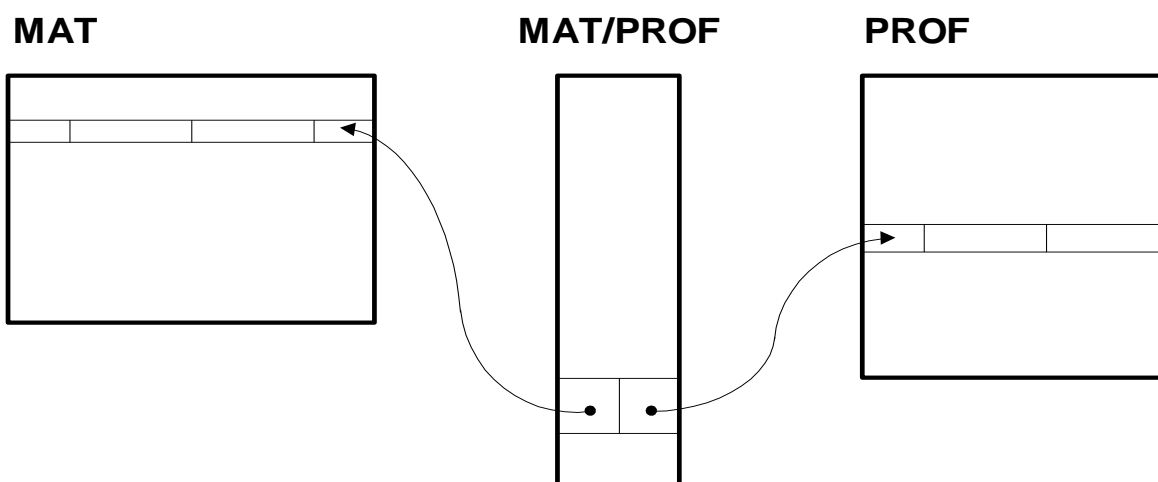
Profesores

Mientras **MAT/EST** relaciona materias con estudiantes, se ve que esto es insuficiente para restaurar toda la información originariamente contenida en el MEP. Como mínimo se necesita una tabla adicional, tal como la **MAT/PROF**. Esta tabla relaciona profesores y materias. En esta oca-

sión existe una entrada para cada materia, y contiene como segundo campo en su fila, un identificador del profesor que dicta la materia.

Las dos relaciones **MAT/EST** y **MAT/PROF** son suficientes para reconstruir el contenido de información de MEP. Puede verificarse que el número total de campos en las cinco sub-relaciones, es mucho menor que el de los campos de MEP. Esta es la fuente del mejoramiento de eficiencia; la misma no se contradice por el hecho de que algunos componentes puedan encontrarse en dos o más relaciones.





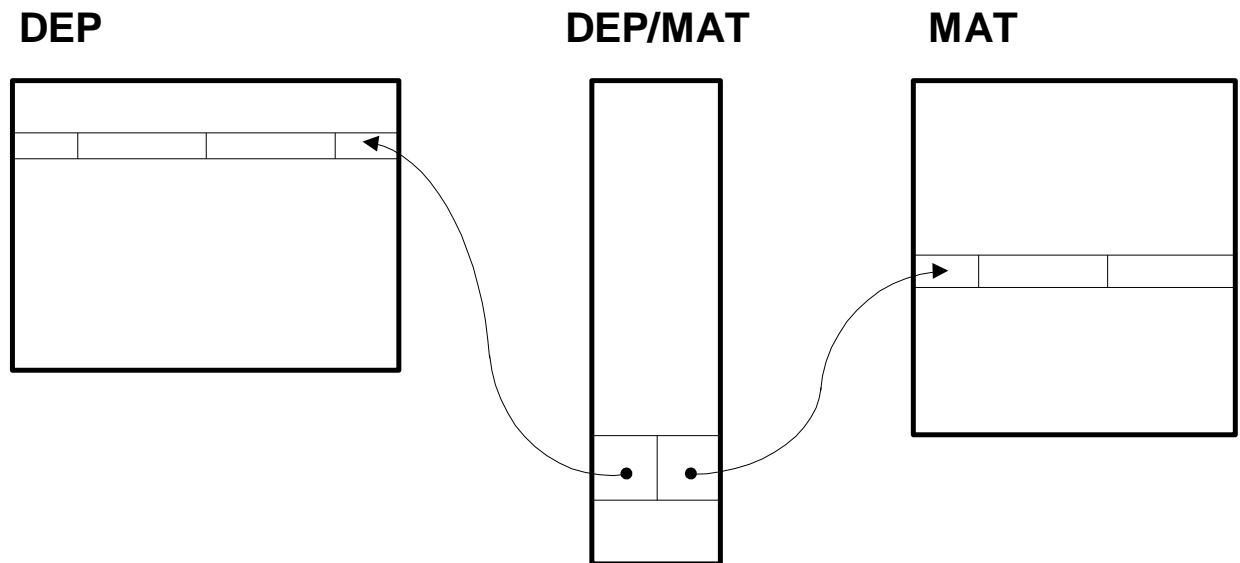
Recuperación

Con **MAT/EST**, **MAT** y **EST**, es posible reconstruir información acerca de cuales estudiantes cursan una determinada materia. Si se proveyeran otros atributos en **MAT/EST**, se podría incluso recuperar registros de asistencia, notas, etc. Si las necesidades se presentan en sentido inverso, por ejemplo si el encargado de las listas necesita encontrar las materias a las que está asistiendo un determinado estudiante, las tablas que se tienen son suficientes para obtener esta información, dado que se puede hacer la búsqueda en la tabla **MAT/EST** y recuperar solamente aquellas tuplas que contienen el número ID de estudiante deseado en el segundo campo. Cada una de estas tuplas, tiene un valor de campo Materia, que señala el nombre de una materia en la cual está registrado el estudiante.

El método para establecer este tipo de requerimientos es simbólico y matemáticamente muy simple. El método para efectuar la búsqueda, puede no ser tan directo, y los procesos que lo logran dentro del motor de la base de datos, dependen del fabricante del software. Por otra parte, la presencia de otra relación, que se llamará **EST/MAT**, que lista primero a los estudiantes y después las materias a las que dicho estudiante concurre (ordenada por ID estudiante), permitirá hallar directamente los datos del estudiante.

Descomposición adicional

La operación original, descompuso la base de datos del ejemplo, en cinco relaciones. Tres de ellas eran descriptivas (entidades) y las otras dos relacionaban a los correspondientes individuos (relaciones). Es posible efectuar una descomposición adicional, que involucra a las materias con los departamentos. Como se ve, el criterio seguido, es análogo al aplicado anteriormente. La situación, puede observarse en el diagrama siguiente.



La tabla de Departamentos, **DEP** lista la información de los departamentos. Se crea entonces otra tabla de relación que vincula cada materia con un departamento determinado (**DEP/MAT**). No disminuye en este caso el tamaño de la tabla de materias, pero sí sus columnas, en las que figuraba la información adicional correspondiente a los departamentos. La cantidad de filas existentes en **DEP/MAT**, es la misma que en la tabla **MAT**.

Depuración de tablas relacionales

Del ejemplo planteado, definitivamente han quedado las siguientes tablas:

1. **ESTUDIANTE** (*Id_estudiante*, Nom_estudiante)
2. **PROFESOR** (*Id_profesor*, Nom_profesor)
3. **MATERIA** (*Id_materia*, Nom_materia)
4. **DEPARTAMENTO** (*Id_departamento*, Nom_departamento)
5. **EST_MAT** (relación entre estudiante y materia)
6. **PROF_MAT** (relación entre profesor y materia)
7. **DEP_MAT** (relación entre departamento y materia)

Entonces, para cada uno de los símbolos utilizados en el modelo conceptual de datos, existirá una tabla en el modelo físico (una por cada entidad y una por cada relación). En la realidad esto no ocurre, ya que no todas las relaciones generan tablas. Su generación depende de la funcionalidad de la relación y de la parcialidad de participación de las entidades en la relación en cuestión. En la enumeración presentada, 5, 6 y 7, representan relaciones. Sus funcionalidades son:

EST_MAT (m,n): ya que un estudiante puede cursar muchas materias y una materia puede ser cursada por muchos estudiantes.

PROF_MAT (1,n): ya que una materia es dictada por uno y solamente un profesor, mientras que un profesor puede dictar varias materias.

DEP_MAT (1,n): ya que una materia pertenece a uno y solamente un departamento, mientras que un departamento tiene varias materias.

Para los dos últimos casos, expresan una dependencia funcional, ya que a partir del conocimiento de la materia, pueden ser determinados el *profesor que la dicta* y el *departamento al que pertenece*. No ocurre esta situación con **EST_MAT**, ya que ni a partir de estudiante o de materia puede determinarse el otro componente de la ocurrencia de la relación. Esta última situación es la que determina la existencia de una TABLA DE RELACIÓN, mientras que en los otros dos casos, tal tabla no es necesaria.

Se plantea ahora la cuestión de cómo poder establecer el vínculo entre materia y profesor por un lado, y por otro, el de materia y departamento. Para ambos casos el razonamiento es el mismo. Al existir una dependencia funcional (cardinalidad 1,1 entre Materia con Profesor y Materia con Departamento), al expresar físicamente esta situación, el identificador de la entidad que se determina a través de la dependencia funcional, pasará como atributo de la entidad condicionante, recibiendo el nombre de Clave Ajena (*foreign key*). De esta forma, la tabla MATERIA quedará conformada por los atributos:

Atributos

- *Id_materia*
- *Nom_materia*
- *Id_profesor*
- *Id_departamento*

Integridad

- Clave primaria: *Id_materia*
- Clave ajena *Id_profesor* **REFERENCIANDO a PROFESOR**
- Clave ajena *Id_departamento* **REFERENCIANDO a DEPARTAMENTO**

Es de notar que el valor de un atributo que se dice que es clave ajena, deberá existir como clave primaria en la tabla a la que referencia, aunque existe la posibilidad de declarar explícitamente que se admita clave ajena nula. Se concluye entonces que en los casos de dependencia funcional, la entidad que condiciona tomará como atributo y clave ajena a la clave de la entidad dependiente.

Considerando el caso de **EST_MAT**, y generalizando, cualquier relación de funcionalidad **m,n** (muchos a muchos), generará una TABLA DE RELACIÓN, cuya clave primaria estará conformada por la concatenación de las claves de las entidades que son colección de la relación, y cada una, será clave ajena referenciando precisamente a las tablas en donde son claves primarias. Así, para el caso de la tabla **EST_MAT**, ésta estará compuesta por los siguientes atributos:

Atributos

- *Id_estudiante*
- *Id_materia*

Integridad

- Clave primaria: *Id_estudiante* + *Id_materia*
- Clave ajena *Id_estudiante* **REFERENCIANDO a ESTUDIANTE**
- Clave ajena *Id_materia* **REFERENCIANDO a MATERIA**

Considerándose la metodología desarrollada en Ingeniería de Software I, los elementos que se transformarán en tablas serán:

- Las entidades
- Las relaciones que sean de funcionalidad **m,n** (muchos a muchos)
- Las relaciones del tipo **0,1 – 0,1**
- Las relaciones que poseen atributos (que necesariamente serán **m,n** o bien **0,1 – 0,1**)
- Las relaciones en las que participen más de 2 entidades
- Para el caso de las jerarquías de clasificación pueden:
 - Generar solamente la tabla correspondiente a la entidad padre
 - Generar la tabla de la entidad padre y tablas para cada uno de sus hijos
 - Generar solamente las tablas de las entidades hijo

LA BASE DE DATOS RELACIONAL

Durante la década de los '80 han aparecido las bases de datos relacionales, las cuales se han hecho más populares que las jerárquicas y las de redes que las precedieron. Con el fin de introducir un cierto orden en la información que surge cada vez a más velocidad sobre las bases de datos relacionales, CODD estableció en 1985 una serie de principios de los cuales al menos seis deben satisfacerse para que una base de datos pueda llamarse totalmente relacional. Estos fueron precedidos de una regla general global, llamada Regla Cero.

Regla 0:

Gestión de una base de datos relacional. Todo sistema que se anuncie como un sistema de gestión de base de datos relacional, debe ser capaz de manejar bases de datos exclusivamente con sus capacidades relacionales.

Regla 1: Representación de la información.

Toda la información de una base de datos relacional, se representa explícitamente en el ámbito lógico y exactamente de una forma: mediante valores en tablas.

Regla 2: Garantía de accesibilidad lógica.

Todos y cada uno de los datos de una base de datos, relacional tienen la garantía de ser accesibles lógicamente mediante el recurso de una combinación de: el nombre de la Tabla, el valor de la clave primaria y el nombre de la columna.

Regla 3: Representación sistemática de la información que falta.

Los valores nulos (que son distintos de la cadena vacía de caracteres o de la cadena de caracteres en blanco, y distintos de cero o de cualquier otro número) tienen la existencia en los sistemas de gestión de bases de datos totalmente relacionales, para representar la información que falta y la información que no es aplicable, de forma sistemática e independiente del tipo de dato.

Regla 4: Sub-lenguaje de datos completo.

Un sistema relacional puede soportar varios lenguajes y varios modos de uso terminal. Sin embargo, debe haber, al menos, un lenguaje cuyas instrucciones puedan expresarse por alguna sintaxis bien definida, como cadenas de caracteres, y que sea completo, soportando todos los términos siguientes:

- Definición de Datos
- Definición de Vistas
- Manejo de Datos
- Limitaciones de integridad
- Autorización o permisos
- Límites de transacción (inicio y fin para hacer permanentes los cambios y deshacer los cambios no permanentes)

Regla 5: Inserción, actualización y borrado de alto nivel.

La capacidad de manejar una relación de base o una relación derivada como un único operador, se aplica no sólo a la recuperación de datos, sino también a la inserción, a la actualización y al borrado de datos.

Regla 6: Independencia de los datos físicos.

Los programas de aplicaciones y las actividades terminales, permanecerán lógicamente inalterados siempre que se realicen cambios en las representaciones de almacenamiento o en los métodos de acceso.

Regla 7: Independencia de los datos lógicos.

Los programas de aplicaciones y las actividades finales permanecerán lógicamente inalterados cuando se llevan a cabo cambios en las tablas de base que conservan la información de cualquier tipo que permita teóricamente su inalterabilidad.

Regla 8: Independencia de la integridad.

Las limitaciones de integridad, específicas de una base de datos en particular, deben ser definibles en un sub-lenguaje de definición de datos y almacenables en el catálogo o diccionario.

Existen otras reglas más esbozadas acerca de los requerimientos de una base de datos relacional, pero las antes expuestas, son las de mayor importancia.

El SQL (Lenguaje de Consulta Estructurado), es un lenguaje estándar universal de manejo de datos, basado en el cálculo aplicado a los predicados y aplicable a las bases de datos relacionales. El hecho de que los desarrolladores supieran de antemano lo que debía ser SQL y lo que requeriría que hiciese, le dio una fuerte base teórica. Esta fue probablemente la primera vez que ocurre en el desarrollo de un lenguaje para computadoras, porque la mayoría de los lenguajes de programación son el resultado de una idea básica que se complementa con una gran cantidad de parches sobre la marcha para resolver los problemas a medida que surgen. Este hecho, el de especificar las necesidades de SQL antes de desarrollar los mecanismos del mismo, dio lugar a un lenguaje elegantemente parsimonioso que consta de relativamente pocos comandos que se pueden utilizar para satisfacer la mayoría de las necesidades de una base de datos muy compleja. Su sencillez hace que SQL sea adecuado tanto para el usuario ocasional, como para el que hace desarrollos profesionales. Se pueden realizar consultas “ad hoc”, y también puede ser embebido en lenguajes de programación.

Definiciones

Teniendo en cuenta el hecho de que SQL es un auxiliar del sistema de gestión de bases de datos relacionales, se definen los siguientes términos dentro del contexto relacional.

Una Tabla es la estructura principal de la base de datos. Es una matriz rectangular con las siguientes propiedades:

1. Es homogénea en sus columnas; en otras palabras, en cualquier columna que se seleccione, los elementos son todos de la misma clase, mientras que los elementos de columnas distintas no tienen porqué ser de la misma clase.
2. Cada elemento, es un único número o una cadena de caracteres – **1FN** - (por lo tanto, al elemento de cualquier fila y columna que se haya especificado, no se encontrará un conjunto de número o un grupo de ellos).
3. Todas las filas de una tabla, deben ser distintas (no se permiten duplicaciones) – **1FN** - .
4. El orden de las filas dentro de una tabla, es indiferente.
5. A las columnas de una tabla, se les asigna nombres distintos, y el orden de las columnas dentro de una tabla, es indiferente.

A estos tipos de tablas, se las llama relación. Si tiene n columnas, entonces se la llama una “relación de grado n”.

Al conjunto de todos los nombres de las columnas de una tabla, se le llama esquema de relaciones (estructura de la tabla), y la Tabla, es una relación sobre el esquema de relaciones. En general, una base de datos, consta de más de una tabla, cada una de las cuales, tiene su propio conjunto de atributos o nombres de columnas. La estructura de una base de datos, es la reunión de todos los esquemas de relaciones de las tablas de la base de datos. Dos esquemas de relaciones distintos, pueden tener algunos nombres de columnas -atributos- en común. Una tabla, es un ejemplo de una relación sobre un esquema de relaciones; es decir, si se cambia de alguna manera los datos de una tabla de relación, se tendrá una relación distinta con el mismo esquema de relaciones. Resumiendo:

- Una relación denotará una *tabla*.
- A un atributo se le llamará *Columna*.
- A un registro único, se le llamará *fila*.
- Al valor individual de la intersección de cualquier fila y columna, se le llamará *dato*.

Las propiedades de las Tablas

Las tablas, deben tener un nombre, y ese nombre no puede ser una palabra clave de SQL. Hay tres tipos de tablas en las bases de datos relacionales: las tablas de base, las tablas virtuales y las tablas temporales. Las tablas temporales se comportan como tablas de base pero presentan restricciones como ser que solamente las puede ver el usuario que la creó. Por otra parte, su vida es efímera ya que son removidas cuando se desconecta la sesión. Una tabla de base, se ajusta a la definición de tabla dada anteriormente. Una tabla virtual, también llamada vista, existe tan sólo

como una definición en el catálogo o diccionario de datos. Cada vez que se accede a una vista, se recupera la definición del diccionario y se ejecuta la definición que hay en la consulta, con lo que se crea una tabla con las columnas que están en el listado de la definición de la vista. A las vistas, se las llama tablas virtuales, porque no existen por derecho propio en la base de datos en la forma en que existen las tablas de base. Por el contrario, la vista se reconstruye a partir de los datos de las tablas de base subyacentes cada vez que se consulta la vista. Mientras que una vista, en su forma más sencilla, puede ser una parte de la tabla base, también puede ser el resultado de unir una parte de una o de varias tablas entre sí. Una vista puede ser también una parte de otra vista, en la que esta última sea una parte de una tabla base.

El valor NULL

SQL incluye el concepto de valor NULL para referirse a la información que es incompleta o de la que no se dispone. Se puede pensar y decir que este valor, es simplemente una instancia más en el dominio de un atributo. Hay distintas opiniones sobre la utilidad o la lógica que subyace en el concepto de los valores NULL. En la actualidad, los valores NULL son en gran medida, una parte del SQL, y en el futuro es probable que aparezcan modificados o adornados, pero nunca eliminados de su sintaxis. Sirven para rellenar un espacio en blanco en una matriz de datos, pero deben utilizarse con precaución. Deberá tenerse en cuenta las siguientes reglas, así como otros usos y precauciones especiales que sobre los NULL se indiquen en los contextos en los que puedan presentarse:

- Un valor NULL de un dato numérico, no es lo mismo que cero.
- Un valor NULL de un dato tipo carácter, no es lo mismo que espacio o blanco.
- Un valor NULL no tiene por qué ser igual a otro valor NULL.
- Un valor NULL no se puede utilizar en una instrucción de proyección (SELECT).
- El tratamiento que de los NULL hacen las funciones agregadas, no es uniforme.