Anales PANEL'81/12 JAIIO
Sociedad Argentina de informática
e Investigación Operativa. Buenos Aires, 1981

MODELO PARA UN SISTEMA GENERALIZADO PARA RECUPERACION DE INFORMACION

Eduardo Luis Miranda
Ramón Falcón 2137, 7º "A", Buenos Aires

OBJETIVO DEL TRABAJO

Presentar las estructuras de datos, y el modo de operación, de un sistema generalizado para la recuperación de información.

ANTECEDENTES

Una aplicación para la recuperación de información, queda definida, con respecto del usuario, por la información que este puede obtener y por la forma en que puede seleccionar dicha información.

El presente estudio trata acerca de este último aspecto, más el problema de la obtención de estadísticas de la información almacenada (ej: $\bar{x}$, s, mín, máx.).

La selección de información se realiza mediante consultas. Estas especifican un criterio a satisfacer y el resultado de su aplicación contra la base de datos es el conjunto de entidades que satisfacen el criterio fijado.

Podemos definir cuatro tipos de consultas:

C_1 - Consultas simples, donde solamente es especificado un valor por medio del operador relacional "=".
Ej. SEXO = FEMENINO, NUMERO-DE-EMPLEADO = 998.

C_2 - Consultas por rangos, mediante el uso de los operadores re-

lacionales"< ", "<= ", " >", ">= ", se especifica un conjunto de valores. Ej. EDAD <= 50.

C_3 - Consultas funcionales, el valor de referencia es función de la información almacenada. Ej. SALARIO MEDIA (SALARIOS).

C_4 - Consultas boolenas, son combinaciones de los tipos precedentes mediante la utilización de los operadores lógicos "Y" "NO" y "O". Ej.EDAD <= 30 Y PROFESION = ANALISTA DE SISTEMAS

Mediante los reemplazos:

- NUMERO-DE-EMPLEADO por NUMERO-DE-IVA.
- EDAD por AÑOS-EN-PLAZA
- SALARIO por CAPITAL
- PROFESION por RAMA-DE-LA-INDUSTRIA,

nos desplazamos de una aplicación de personal hacia una de información empresaria y los ejemplos mencionados continuan teniendo sentido.

De lo expuesto, bajo C_1, C_2, C_3 y C_4 las consultas a realizarse, sobre una o distintas aplicaciones difieren únicamente en el o los nombres de atributos que caracterizan a las entidades de la aplicación.

Todo lo antedicho se puede extender al problema de las estadisticas.

Por lo tanto desarrollando algoritmos y estructuras, haciendo abstracción de dichos nombres, se puede construir un sistema apto para múltiples aplicaciones.

Este enfoque presenta desde el punto de vista del usuario las siguientes ventajas:

- desaparece la distinción entre consultas programadas y espontáneas.
- se logra para todo tipo de consultas, de una o distintas aplicaciones, una interfase única, lo que simplifica la explotación de información.

respecto del sector de sistemas, las características más destacables son:

- programación única, esto implica un menor costo de mantenimiento.
- uniformidad en los procedimientos de seguridad.
- simplicidad en los procedimientos operativos.

DEFINICIONES PRELIMINARES

Todo elemento almacenado tiene asignado uno o dos números positivos, denominados dirección, que, indican, donde en un espacio de almacenamiento dado, se encuentra registrado. En el caso de que los números sean dos, tienen la forma (P,L), donde P indica página y L Línea.

La página es la unidad de transferencia entre la memoria central y un soporte externo. La línea es la unidad de tratamiento lógico; existiendo k líneas por página.

Directorio es un conjunto de índices, donde cada uno de estos puede ser visto como una colección de pares de la forma (valor, dirección). Si el valor determina univocamente a una entidad,entonces la dirección es la de esa entidad, en caso contrario, la

dirección apunta al primer elemento de una lista de direcciones de las entidades que poseen dicho valor para un atributo dado.

Diccionario es el conjunto de nombres de atributos, descriptores y apuntadores, utilizado en la interpretación almacenada.

Cada elemento del diccionario tiene la forma (atributo, descriptor, dirección) donde la dirección apunta a un índice del directorio, que es el correspondiente a los valores que ha tomado el atributo para las entidades existentes en la base de datos.

Los atributos se agrupan en una de dos categorías:

- Atributos básicos, son aquellos que están presentes para todas las entidades.
- Atributos variables, son los que pueden estar presentes o no dependiendo de la entidad que se considere.

En una aplicación de personal, son ejemplos del primer tipo el APELLIDO y el NUMERO-DE-EMPLEADO en tanto que del segundo lo son NOMBRE-DE-LA-ESPOSA y TITULO-UNIVERSITARIO.

PRINCIPIOS DE OPERACION

Sea una m-upla $(E_1, E_2, \ldots, E_m)$, cuyos elementos llamados entidades, están caracterizados por los atributos a_i, $1 \leq i \leq n$ y $1 \leq j \leq m$, estos atributos pueden asumir uno de dos valores y sea $D^{n \times m}$ una matriz, cuyos elementos $d_{ij} = 1$ si la entidad j posee el atributo i y $d_{ij} = 0$ si no lo posee (ver fig. 1).

$(E_1, E_2, \ldots, E_m) =$

$((a_1, a_2, \ldots, a_n)_1, (a_1, \ldots, a_{k-1}, a_{k+1}, \ldots, a_n)_2, \ldots, (a_2, \ldots, a_{n-1}, a_n)_m)$

	1	2		m
a_1	1	1		0
a_2	1			1
a_{k-1}		1		
a_k		0		
a_{k+1}		1		
a_{n-1}				1
a_n	1	1		1

$D^{n \times m}$

FIG. 1

Para hallar todas las entidades que cumplen con la condición de poseer el atributo k, $1 \leq k \leq n$, es necesario observar la fila k de la matriz D.

Las entidades que satisfacen la condición son aquellas, tales que $d_{ij} = 1$, en ese caso se utiliza j como la dirección de E (ver fig. 2a).

Para resolver consultas boolenas se recuperan las filas necesarias y se opera sobre ellas (ver fig. 2b y 2c).

$$q_k = (1, 0, 0, 0, 1, \ldots, 1, 0) \in D^{n \times m}$$

Las entidades E_1, E_5, ... y E_{m-1} cumplen con la condición de poseer. q_k.

a

$$q_k = (1, 0, 0, 0, 1, \ldots, 1, 0)$$
$$q_g = (1, 1, 0, 0, 0, \ldots, 1, 1)$$
$$q_k \wedge q_g = (1, 0, 0, 0, 0, \ldots, 1, 0)$$

Las entidades E_1 y E_{m-1} cumplen con la condición de poseer. $q_k \wedge q_g$.

b

$$q_k = (1, 0, 0, 0, 1, \ldots, 1, 0)$$
$$q_g = (1, 1, 0, 0, 0, \ldots, 1, 1)$$
$$q_k \vee q_g = (1, 1, 0, 0, 1, \ldots, 1, 1)$$

Las entidades E_1, E_2, E_5, ..., E_{m-1}, E_m cumplen con la condición de poseer. $q_k \vee q_g$.

c

FIG. 1

Para extender el modelo a atributos multivaluados con grado mayor que dos, es necesario modificar la definición de la matriz D. En este caso cada fila corresponde a un par (a_i, V_k), siendo $d_{ij} = 1$ si la entidad j posee el atributo i con valor k.

Este cambio en la definición no afecta la forma de las operaciones.

Las consultas por rango, se resuelven mediante series de "o".

El espacio físico ocupado por D sería desmesurado de no ser por el hecho de que una gran cantidad de d_{ij} son ceros. Recurriendo a las técnicas para el manejo de matrices ralas se soluciona el problema tanto de espacio como de tiempo de exploración.

Una de estas técnicas consiste en particionar D en submatrices fila, almacenando únicamente aquellas que poseen uno o más elementos no nulos. Todas las submatrices de una misma fila dan origen a una lista de direcciónes, salvo en el caso de determinación unívoca (ver definición preliminares).

La cantidad de elementos que contiene cada una de estas submatrices es fijo y está relacionado con el tamaño de página del espacio de datos básicos.

Si a_1 , a_2 , ..., a_k son atributos básicos, ellos son almacenados en el espacio de datos básicos y los a_{k+1} , a_{k+2} , ..., a_n atributos variables son almacenados en el espacio de datos variables (ver fig. 3).

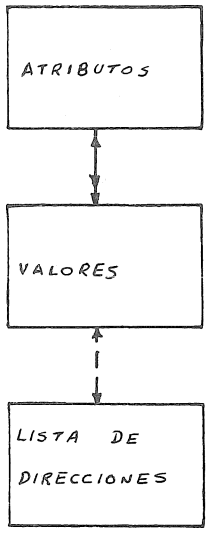
FIG. 3

ESTRUCTURAS DE DATOS

DICCIONARIO / DIRECTORIO

- Funciones: Proveer el registro de los atributos y valores almacenados, de forma tal que permita:
  - Independencia de datos.
  - Localizar rápidamente el conjunto de entidades que satisfacen una consulta dada.

- Estructura: Coexisten en el mismo espacio de almacenamiento tres clases de estructuras (ver fig. 4 y 6):
  - Arbol de atributos
  - Arbol de valores
  - Lista de direcciones



ESQUEMA LOGICO DEL DICCIONARIO/DIRECTORIO

(LA LISTA DE DIRECCIONES, PUEDE EXISTIR O NO DEPENDIENDO DEL ATRIBUTO)

FIG. 4

Características de los árboles de atributos y valores.

Ambos árboles son del tipo B de orden m, y gozan de las siguientes propiedades:

- Todo nodo tiene a lo sumo m hijos
- Todo nodo excepto la raiz y las hojas tienen por lo menos $\lceil m / 2 \rceil$ hijos
- La raíz tiene por lo menos dos hijos (a menos que sea una hoja)
- Todas las hojas aparecen al mismo nivel y no poseen formación (por lo tanto se los puede representar por el apuntador nulo).
- Un nodo que no es hoja, con k hijos, contiene k - 1 claves.

Estructura interna de cada nodo

Arbol de atributos

$(n, D_0, (A_1, D_1, I_1), (A_2, D_2, I_2), \ldots, (A_n, D_n, I_n))$

donde

- $n < m$, cantidad de ternas (A_i, D_i, I_i) en el nodo
- D_i, $0 \leq i \leq n$, son apuntadores a los subárboles del nodo
  A_i, $1 \leq i \leq n$, son los nombres de atributos, y los
  I_i, $1 \leq i \leq n$, son los descriptores de A_i y el puntero al árbol de valores que corresponde al atributo A_i.
- $K_i < K_{i+1}$, $1 \leq i < n$
- Todos los nombres de atributos en el subárbol D_i son mayores (lexicográficamente) que A_i.
- Los subárboles D_i, $0 \leq i \leq n$, cumplen las propiedades anteriores.

Arbol de valores

$(n, D_0, (V_1, D_1, H_1), (V_2, D_2, H_2), \ldots, (V_n, D_n, H_n))$

Se aplican las mismas definiciones que para el árbol de atributo con las siguientes excepciones:
- V_i, es uno de los valores que ha tomado el atributo.
- H_i, es un puntero, que contiene la dirección de una entidad dada, si esta queda univocamente determinada por el par atributo valor, y sino, contiene la dirección del primer elemento de una lista de direcciones. En cualquiera de los dos casos la forma de H_i es (P,L).

Comportamiento de las estructuras (ambos árboles)

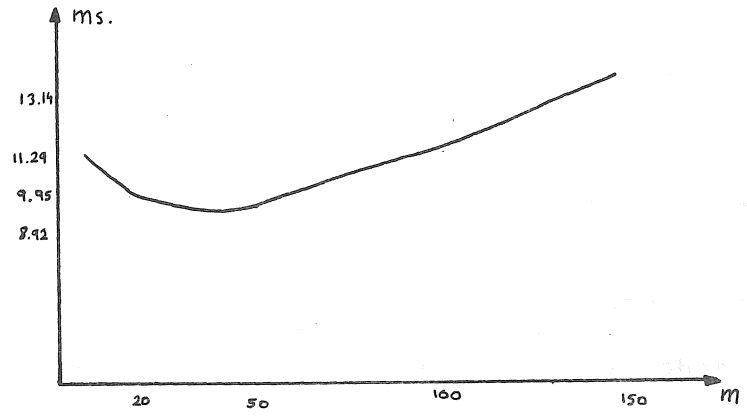
Suponiendo que una vez recuperado el nodo, se realiza una búsqueda binaria sobre las claves, ya sean valores o atributos, el tiempo máximo de búsqueda T para encontrar un tributo o valor es:

$$T \cong (\log_2 (n + 1) / 2) \left(\frac{t_r + t_p}{\log_2 m} + \frac{a \cdot m}{\log_2 m} \right)$$

donde N es la cantidad de claves (atributos o valores de un atributo).

t_r = demora rotacional (promedio)
t_p = tiempo de posicionamiento de cabezas (promedio)
a = $(\alpha + \beta + \gamma)$ tc
t_c = tiempo de transferencia por cada carácter
α, β, γ son la longitud expresadas en caracteres de D_i, A_i o V_i y I_i o H_i según el caso.

Graficando la parte de la expresión que depende de m obtenemos la siguiente curva (ver fig. 5)



$t_r + t_p = 35$ ms

$\alpha + \beta + \gamma = 40$ CARACTERES

$t_c = 10^{-5}$ sec

Se observa que existe un márgen amplio para la elección de m, por lo tanto dentro de ciertos límites, m se fijará teniendo en cuenta las características de la información sobre la cual se va a operar, y la memoria disponible para buffers.

El número máximo de accesos a disco para este tipo de estructuras es:

$$N_a \leq \log_{\lceil m/2 \rceil} (N + 1) / 2 + 1 \qquad (1)$$

Lista de direcciones

Cuando un par de atributo valor no determina univocamente a una entidad, el número de éstas que satisfacen dicho par es desconocido, es necesario entonces adoptar una estructura que provea gran flexibilidad en su capacidad de almacenamiento.

Estructura interna de un elemento de lista.

(D, B, A)

donde:

D, es el número de página de una entidad que satisface el par atributo valor dado. Conceptualmente se lo puede asociar con el subíndice de las submatrices fila mencionadas en Principios de operación.

B, es una cadena de k bits, b_1, b_2, ..., b_k, siendo b_j = 1, $1 \leq j \leq k$, cuando la entidad que satisface el par se encuentra en la línea j de la página D y cero en otro caso. k es igual al número de líneas por páginas existentes en el espacio de datos básicos.

A, es el apuntador al próximo elemento de lista, siendo su forma (P, L).

Para que exista un elemento de lista, correspondiente a la página D, debe existir en esa página por lo menos una entidad que satisfaga el par atributo valor dado.

La lista se mantiene ordenada en forma ascendente respecto de D.

Estrategia de almacenamiento.

Los atributos pueden clasificarse en densos y poco densos.

Son densos cuando sus valores son comunes a un gran número de entidades, este es el caso de atributo SEXO en una aplica-

ción del personal; mientras que el atributo CARGO-QUE-DESEMPEÑA, es poco denso, pues son pocas las personas que desempeñan la misma función.

Para la primera categoría de atributos, es conveniente agrupar todos los elementos de una o más paginas asignadas en forma exclusiva, a los efectos de reducir el número de accesos a un soporte externo.

Para la segunda categoría, esta política se traduce en desaprovechamiento del espacio de almacenamiento; por esto, cuando un atributo valor dado, comparten el espacio de una página con los elementos de otras listas correspondientes también a atributos poco densos.

Número de accesos a disco

Para los atributos densos es:

$$N = r / (l \times f) \qquad (2)$$

donde

- r , es el número de entidades que satisfafe un par dado
- l , es el número de líneas por página del diccionario/directorio
- f , es el número medio de entidades que son referenciadas por un elemento de lista.

ESPACIO DE DATOS BASICOS

Función: Almacenar los atributos básicos

Estructura interna de un vector de atributos básicos

$$(V_1 , V_2 , \ldots , V_n , S , T)$$

donde

- V_i , es el valor asociado con el atributo A_i
- S , es el apuntador al primer elemento de la lista de atributos variables
- T , indica con que imagen ha de interpretarse el primer elemento de la lista de datos variables, esto último será explicado mas adelante.

Estrategia de almacenamiento

Los vectores de atributos básicos se agrupan en páginas, existiendo k de estos vectores (línea) por página.

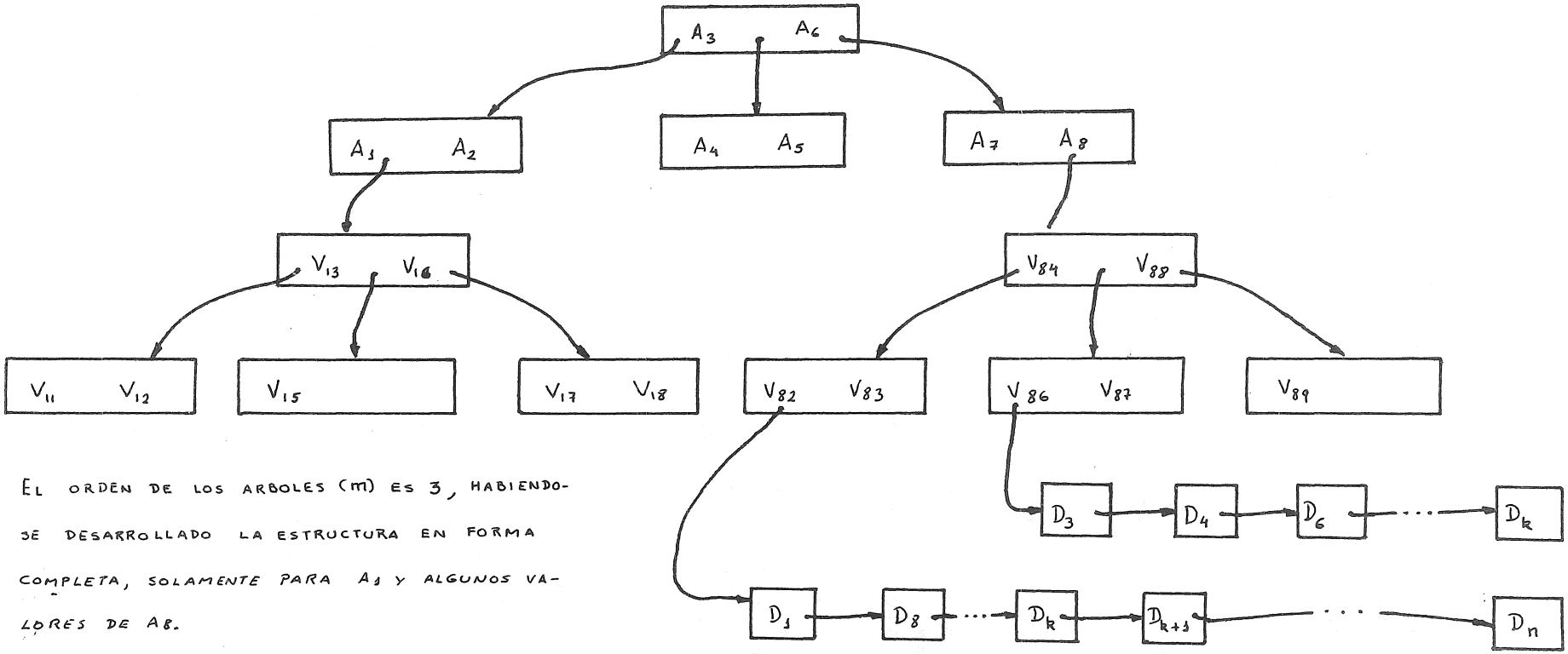
Método de acceso

Los vectores de atributos básicos son accedidos a través de las direcciones (P,L) existentes en el diccionario.

Interpretación de los datos almacenados

Mediante las descripciones existentes en el diccionario

ESTRUCTURA DEL DICCIONARIO



EL ORDEN DE LOS ARBOLES (m) ES 3, HABIENDOSE DESARROLLADO LA ESTRUCTURA EN FORMA COMPLETA, SOLAMENTE PARA A_1 Y ALGUNOS VALORES DE A_8.

A_8, ES UN ATRIBUTO, CUYOS VALORES DETERMINAN UNIVOCAMENTE A LAS ENTIDADES ASOCIADAS, POR LO TANTO, PARA SUS VALORES NO EXISTEN LISTAS DE DIRECCIONES, LO CONTRARIO OCURRE CON A_8.

FIG. 6

DATOS VARIABLES

Función: Almacenar los atributos variables

Estructura: Los atributos variables, se organizan en forma de lista ordenada, respecto del código del atributo.

Estructura interna de un elemento de la lista de atributos variables.

(X, V, S, T)

donde

X, es un código de atributo, mediante él se establece la vinculación entre el valor V y su significado y descripción, almacenados en el diccionario.

S, T, se aplican las mismas definiciones que en el caso de atributos básicos.

Estrategia de almacenamiento.

Los elementos de una lista de atributos variables se agrupan a nivel de página, para minimizar el número de accesos a dispositivos de almacenamiento externo.

Puesto que los valores que toman los distintos atributos pueden requerir distintas capacidades de almacenamiento, el espacio de una página es múltiplemente definido, seleccionándose una definición (imagen) en función de la cantidad de caracteres que necesita un valor para ser almacenado.

En el campo T se indica cual es la imagen a utilizar para recuperar el próximo elemento de la lista.

Cuando la capacidad de una página es excedida, se habilita una nueva página aplicándose a ella las definiciones anteriores.

PROCEDIMIENTOS PARA LA OPERACION DEL SISTEMA

Recuperación de información.

Dada una consulta, se determina mediante el uso del diccionario / directorio, que entidades satisfacen el criterio de recuperación especificado.

Sea por ejemplo una consulta de tipo 1. Para este caso se procede a buscar en el árbol de atributos, aquel por el cual se ha preguntado. Una vez hallado dicho atributo, mediante su información asociada I_i (descriptor y puntero a la raíz del árbol de valores), accedemos al árbol de valores correspondiente. En dicho árbol se busca el valor que satisface la condición especificada.

Se presenta aquí dos casos:

- El par atributo valor determina univocamente a una entidad luego H_i (ver árbol de valores), es la dirección de la entidad buscada.
- El par atributo valor determina un conjunto de entidades.
  En este caso H_i es la dirección del primer elemento de una lista de direcciones. Esta lista es recorrida ele-

mento a elemento, utilizándose el siguiente procedimiento para generar las direcciones de las entidades correspondientes:

- D, es el número de página del espacio de datos básicos, en que una de las entidades que satisface el par, está almacenada.
- Recorriendo luego la cadena de bits B, y para cada $b_i = 1$ se genera la dirección (D, i), donde i es un número de línea dentro de la página D.

Mediante las direcciones obtenidas (no importa el caso), se accede al vector de atributos básicos de la entidad correspondiente y luego a traves del puntero S y de acuerdo con la imagen indicada por T a la lista de atributos variables.

Mediante un proceso de edición se elaboran los datos para presentarlos en forma adecuada y se emite una respuesta. Este proceso se repite hasta agotar la lista de direcciones.

Para distintos tipos de consulta, solo varía la forma en que los árboles son recorridos y la necesidad de incorporar para las consultas de tipo 3 y 4 un procedimiento de cálculo y uno de operaciones lógicas. Una vez desarrollado el procedimiento genérico para cada tipo de consulta se habrá obtenido un procesador de consultas de uso general.

Incorporación de una entidad a la base de datos

Como resultado de este proceso, los atributos de una nueva entidad son incorporados a los espacios de datos y el diccionario / directorio es actualizado.

Se determina en primer lugar, donde, en el espacio de datos básicos ha de ser almacenado el vector correspondiente a la nueva entidad.

Para la actualización del diccionario / directorio, se han de considerar los siguientes casos:

- El par atributo valor considerado determina univocamente a una entidad.
- El par atributo valor, no determina univocamente a una entidad y es además denso.
- El par atributo valor, no determina univocamente a una entidad y es poco denso.

Para cada atributo valor se buscará primero en el árbol, de atributos, el atributo considerado, luego se accederá al arbol de valores buscándose el valor especificado. Dicho valor puede hallarse o no. Para esta última alternativa, el valor es incorporado al árbol colocándose en H_i la dirección de la entidad en el espacio de datos básicos o bien la dirección del primer elemento de la lista de direcciones asociadas, en este caso debemos proceder a crear el primer elemento de la lista.

Si el valor hubiese sido hallado, y el par determinara a la entidad en forma univoca, se presenta una condición de error, si el par no determina a una entidad univocamente, entonces se procede a actualizar la lista de direcciones.

En la actualización de la lista se presentan dos situaciones, se incorpora un nuevo elemento a la lista, o se "enciende" un bit

en la cadena B de un elemento ya existente. Esto ocurre, cuando en la página donde va a almacenar el vector de atributos básicos de la nueva entidad, ya existe otra entidad que satisface el par.

La característica de denso, o poco denso influye únicamente en la forma de seleccionar la página del diccionario / directorio en que ha de ser colocado un nuevo elemento de lista.

En todos los casos se deben respetar las características de los árboles.

Luego se crea el vector de atributos básicos y se almacena en el lugar previamente determinado. De igual forma se procede con la lista de atributos variables, determinando para cada uno de sus elementos la imagen a utilizar en el almacenamiento y encadenando los elementos en forma ordenada respecto del código de atributo.

El proceso se repite para todos los pares atributo valor que caracterizan a la entidad a incorporar.

Incorporación de la definición de un nuevo atributo en una base de datos operativa.

Como resultado de esta operación, la definición de las entidades que componen la base de datos se ve modificada por el agregado de un nuevo atributo.

Esta nueva definición no afecta los algoritmos presentados, pues ellos operan, como se ha visto, sobre las definiciones existentes en el diccionario, alcanzándose de esta forma la independencia de datos.

El procedimiento de incorporación consiste simplemente en la actualización del árbol de atributos, mediante la incorporación del nuevo nombre de atributo y su descripción asociada. En esta incorporación deben respetarse las condiciones de existencia de un árbol tipo B.

Obtención de estadísticas

Para la obtención de estadísticas, se localiza, mediante el árbol de atriburos, el árbol de valores correspondiente, y se lo recorre contabilizando los datos necesarios para la estadística requerida.

ANALISIS DEL COMPORTAMIENTO DEL SISTEMA EN TERMINOS DEL NUMERO DE ACCESOS NECESARIOS PARA RESPONDER A UNA CONSULTA.

El número total de accesos, está dado, por la suma de los accesos al diccionario / directorio y a los espacios de datos.

El número de accesos a los espacios de datos es fácilmente calculable, pues usualmente se requiere un acceso a una página del espacio de datos básicos y uno o más accesos a las páginas del espacio de datos variables para recuperar la lista de atributos variables de la entidad correspondiente.

$$\text{Accesos a datos} = \bar{A}_b + \bar{A}_v \leq 1 + \bar{A}_v \leq 2^{(*)}$$

(*) si el tamaño de página es adecuado.

El número de accesos al diccionario / directorio, se ve afectado por un gran número de parámetros que intervienen en el análisis. Estos son:

- Tipo de consulta
- Orden de los árboles de atributo y de valor
- Número de entidades en la base de datos
- Características del par atributo valor al que se hace referencia
- Cantidad de atributos definidos
- Líneas por página de las listas de direcciones
- Número medio de entidades a las que hace referencia cada elemento de la lista de direcciones

De acuerdo con todo ésto, el estudio se limitará al trazado de curvas (ver fig.9), para valores dados (ver apéndice).

El elevado número de accesos correspondientes a la consulta tipo II, en relación con las tipo I y IV, puede ser corregido aumentando el número medio de entidades a las que hace referencia un elemento de una lista de direcciones. Esto se logra mediante la asignación de direcciones a los vectores de atributos básicos de acuerdo con un criterio de agrupamiento(clustering), basado en aquellos atributos sobre los cuales se van a efectuar consultas por rango.

ESTRUCTURA DE LOS ESPACIOS DE DATOS BASICOS Y VARIABLES

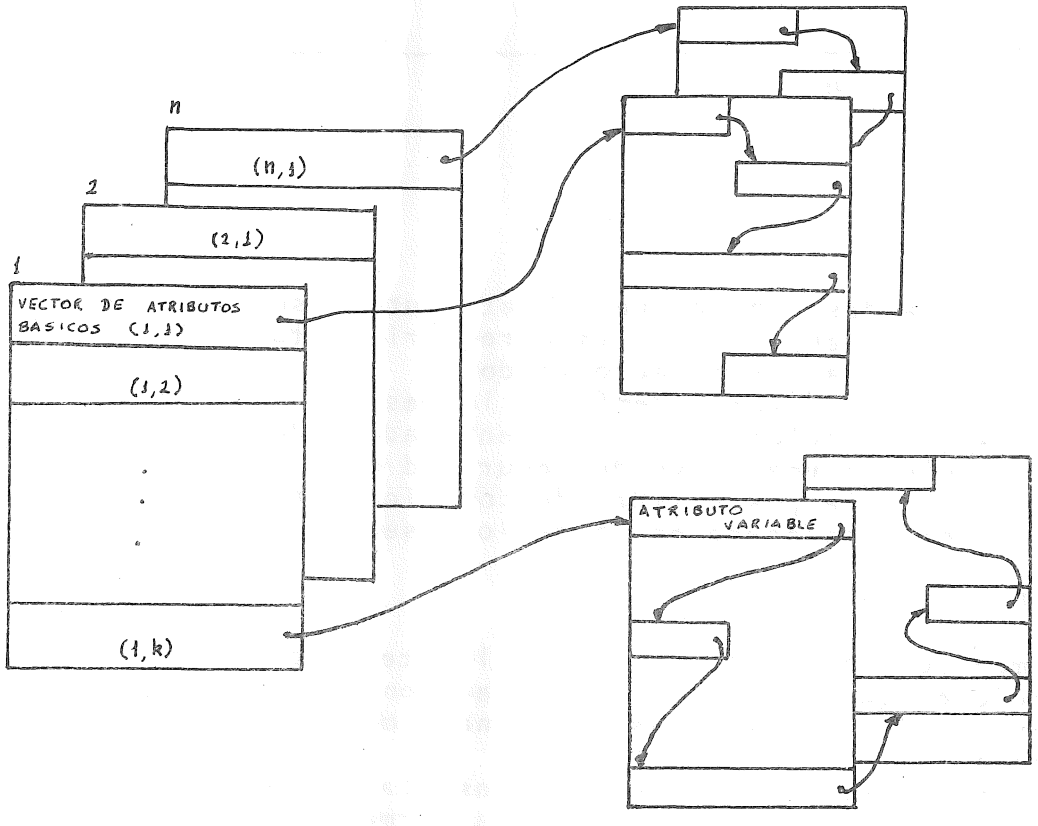


FIG. 8

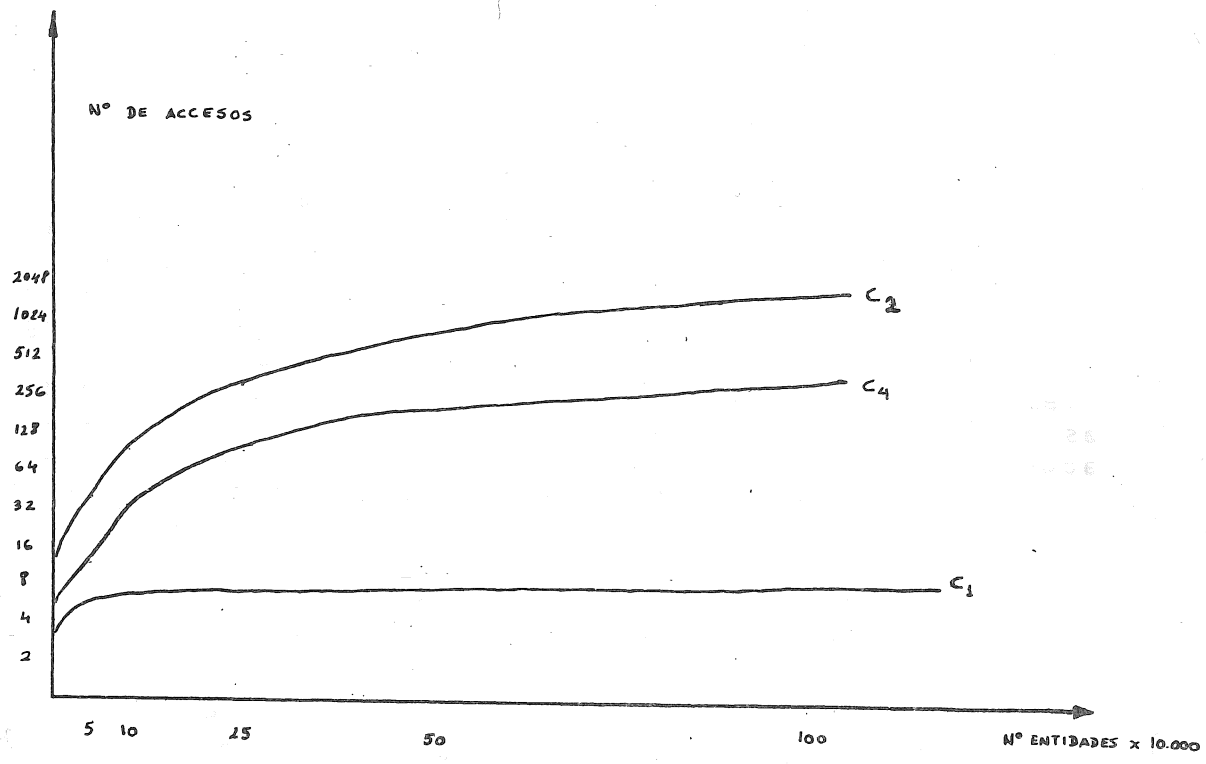


Fig. 9

SEGURIDAD

Junto a la descripción de los atributos, se deben incorporar claves que impidan la consulta o requerimiento de información por parte de usuarios no autorizados.

Se ha de considerar, además de la restricción a nivel de atributos, la restricción a nivel de entidades, debiendo incorporarse a este efecto una clave de seguridad en el vector de atributos básicos; y la posibilidad de codificar la información si esta ha de ser almacenada en un medio inseguro.

AREAS DE APLICACION

El modelo propuesto, es especialmente apto para aquellas organizaciones que necesitan consultar, en forma selectiva extensas bases de datos, y satisfacen una o más de las siguientes condiciones:

- no existen criterios de consulta predominante
- existen varias aplicaciones formalmente iguales
- la definición de las entidades almacenadas, evoluciona constantemente mediante el agregado y / o eliminación de atributos.

ADMINISTRACION DE ESPACIO

La política de administración de espacio, es función de la forma en que se determinan las direcciones de las entidades (vector de atributos básicos) almacenadas en la base de datos

Si no se aplica respecto de las entidades, ningún criterio de agrupamiento, el manejo de espacio puede basarse en listas de espacios recuperados y espacios nunca utilizados.

Esta técnica se implimenta para todas las estructuras del sistema con las siguientes particularidades:

Diccionario:

Una lista para aquellas páginas que contienen al menos una línea disponible y la página en cuestión es utilizada para almacenar los elementos de la lista de direcciones de atributos poco densos.

Dentro de cada página las líneas disponibles son marcadas mediante valores especiales.

Una lista de aquellas páginas que habiendo sido usadas, actualmente no contienen ninguna línea activa.

Una página que contiene una o más líneas (pero no todas) disponibles, y que es utilizada para almacenar los elementos de una lista de direcciones correspondientes a un atributo denso, no pertenece a ninguna lista de espacio recuperado, aunque sus líneas son marcadas como disponibles para ser vueltas a usar por elementos de la lista de direcciones correspondientes al par atributo valor dado.

Este mecanismo asegura el uso de estas páginas en forma exclusiva.

Espacio de datos básicos:

Existe una lista que vincula las páginas que poseen líneas disponibles, dentro de cada página dichas líneas son marcadas.

Espacio de datos variables:

Existe una lista a la cual pertenecen todas aquellas páginas que habiendo sido usadas ya no lo son.

Una página que contiene atributos variables de una entidad, aunque no esté completamente ocupada, no pertenece a ninguna lista, ya que las páginas son de uso exclusivo de una entidad.

Dentro de la página los espacios libres son marcados.

Cuando se aplica un criterio de agrupamiento, el manejo del espacio, es parte principal de la implementación de dicho criterio. Por tal razón no es tratado aquí.

IMPLEMENTACION PARA ATENDER CONSULTAS RESPECTO DE DIFERENTES TIPOS DE ENTIDADES

Lo más indicado para sortear este problema, es introducir un nivel adicional en el diccionario / directorio (ver fig. 10).

Mediante este nuevo nivel, se determinará, el árbol de atributos sobre el cual se ha de operar y luego se procederá en la forma anteriormente descripta.

Otro tipo de solución, menos elegante, que la expuesta consiste en generar múltiples instancias del sistema (ej. catalogar programas y archivos bajo distintos nombres), una instancia para cada aplicación.

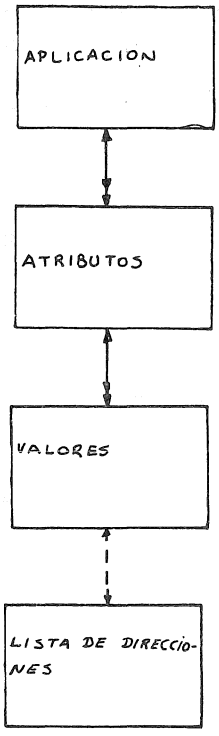


FIG 10

APENDICE

En el cálculo de los valores graficados se han aplicado las expresiones (1) y (2), habiéndose asumido los siguientes valores:

- m: orden de los árboles = 120
- A: Cantidad de atributos definidos = 100
- l: Líneas por página, cuando se almacena un elemento de una lista de direcciones = 200
- k: líneas por página del espacio de datos básicos = 32

En el caso de las consultas II y IV, se aplica la siguiente tabla: (ver fig. 11)

Curva I, consulta simple respecto de un par que determina en forma univoca a una entidad.

R_1 = accesos para localizar el atributo

$\leq \log_{\lceil m/2 \rceil} (A + 1) / 2 + 1$

$\leq \log_{60} 50,5 + 1 \leq 1,96$

R_2 = accesos necesarios para localizar el valor

$\log_{\lceil m/2 \rceil} (N + 1) / 2 + 1$

$\log_{60} (N + 1) / 2 + 1$

$R : R_1 + R_2 = 2,96 + \log_{60} (N + 1) / 2$

Curva II, consulta por rango

$R_1 \leq 1,96$

$R_2 \leq \log_{60} (v + 1) / 2 + 1$

R_3 = número medio de accesos originados en el recorrido de una lista de direcciones $\leq r / (1 * f)$

Respecto de esta consulta, se supone además que se resuelve mediante una serie de operaciones "o", y el valor de referencia es la mediana del conjunto de valores.

$$R \leq R_1 + R_2 + \frac{(v + 1)}{2} R_3 \leq 2,96 + \log_{60} \frac{(v + 1)}{2} + \frac{(v + 1)}{2} \quad \frac{r}{200 f}$$

Curva IV, consulta boolena de la forma $A_1 = V_1$ y $A_2 = V_2$

$R_1 \leq 1,96$

$$R_2 \leq \log_{60} \frac{(v + 1)}{2} + 1$$

$R \leq r / (1 * f)$

Para resolver este tipo de consulta, es menester buscar dos atributos, dos valores y recorrer dos listas de direcciones, por lo tanto:

$$R = 2 \left(1,96 + \log_{60} \frac{(v + 1)}{2} + 1 + r / 200 f\right)$$

$$= 5,92 + 2 \log_{60} \frac{(v + 1)}{2} + r / 100 f$$

Número de entidades en la base de datos	Número de valores que toma el atributo	Número de entidades que poseen el valor	Número medio de entidades que un elemento de lista referencia *
(N)	(v)	(r)	(f)
50.000	8	5.000	3,2
100.000	10	6.250	2,0
250.000	13	12.500	1,60
500.000	15	23.400	1,50
1.000.000	17	41.600	1,33

Fig 11

* Suponiendo una distribución uniforme de las entidades que contienen un valor dado, a través de toda la base de datos,

$$f = (r * 32) / N$$

BIBLIOGRAFIA

- Fundamentals of Data Structures. Horowitz - Sahni. Computer Science Press.
- The Art of Computer Programming , Vol. I y III. Donald Knuth. Addison Wesley.
- Systems Programming. David Hsiao. Addison Wesley.
- Organización de las Bases de Datos. James Martin. Prentice - Hall.
- The Design and Analysis of Computer Algorithms. Aho - Hopocrof - Ullman. Addison Wesley.
- Diseño de Programas para Sistemas. Gauthier - Ponto. Paraninfo.
- Data Structures. Berztiss. Academic Press.