

DISEÑO DE BASES DE DATOS DISTRIBUIDAS

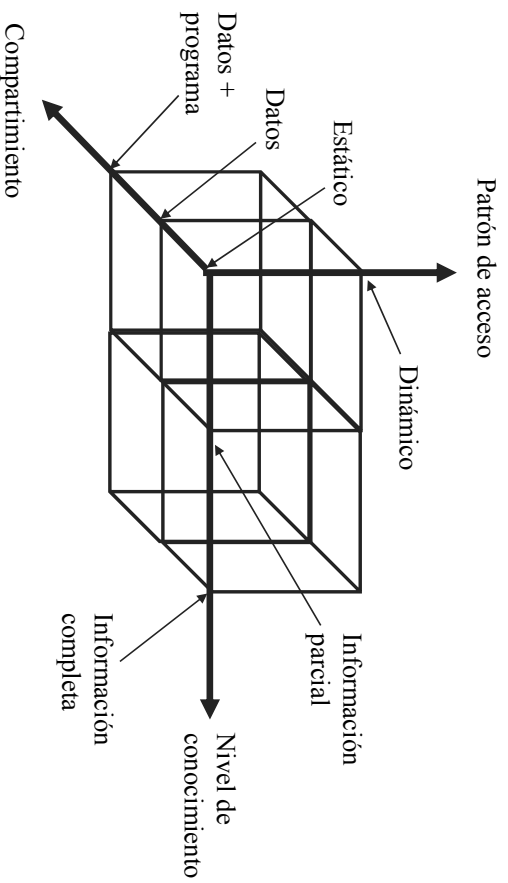
M.C. Víctor J. Sosa Sosa

Problema de Diseño

- En un marco general
 - ◆ tomar decisiones acerca de la ubicación de los datos y los programas, en los sitios de una red de computadoras, así como también posiblemente el diseño de la propia red.
- En los SABDDs, la ubicación de las aplicaciones supone
 - ◆ ubicación del software de los SABDDs, y
 - ◆ ubicación de las aplicaciones que corren sobre la base de datos.

M.C. Víctor J. Sosa Sosa

Dimensiones del Problema



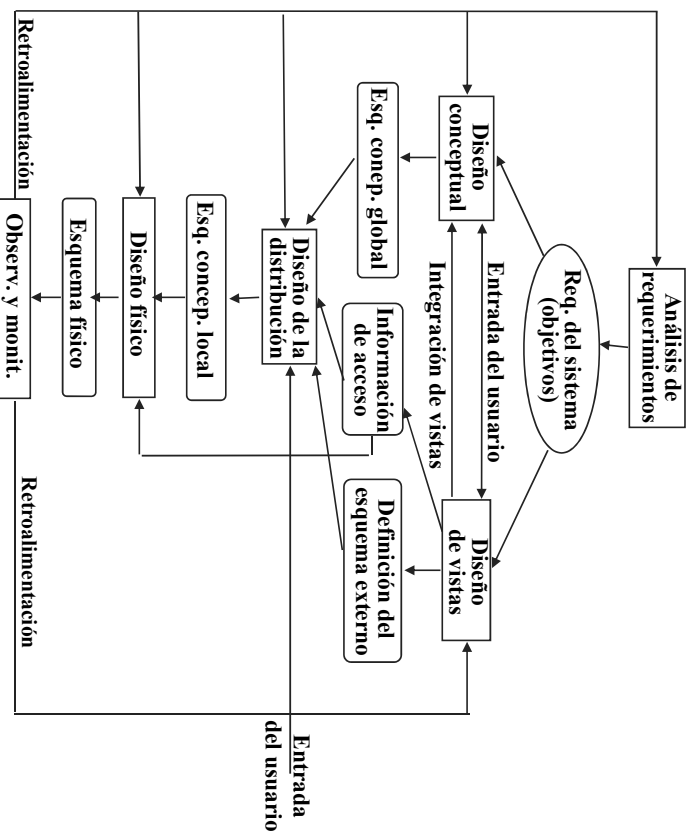
M.C. Víctor J. Sosa Sosa

Diseño de la Distribución

- ❑ Descendente.
 - ◆ Principalmente en sistemas que inician de cero.
 - ◆ Principalmente en sistemas homogéneos.
- ❑ Ascendente.
 - ◆ Cuando las bases de datos ya existen en los diferentes sitios.

M.C. Víctor J. Sosa Sosa

Diseño Descendente



M.C. Víctor J. Sosa Sosa

Problemática del Diseño de la Distribución

- ❑ ¿Por qué fragmentar todo?
- ❑ ¿Cómo fragmentar?
- ❑ ¿Cuánto fragmentar?
- ❑ ¿Cómo probar la corrección?
- ❑ ¿Cómo ubicar los fragmentos?
- ❑ ¿Qué requerimientos de información?

M.C. Víctor J. Sosa Sosa

Fragmentación

- ❑ ¿Podemos distribuir sólo relaciones?
- ❑ ¿Cuál es la unidad razonable de distribución?
 - ◆ Relación.
 - Las vistas son subconjuntos de relaciones (localmente).
 - Comunicación extra.
 - ◆ Fragmento de relación (sub-relación).
 - Ejecución concurrente de un número de transacciones que accedan diferentes porciones de una relación.
 - Las vistas que no puedan ser definidas sobre un solo fragmento requerirían procesamiento extra.
 - El control de la semántica de datos (especialmente forzar la integridad) es más complejo.

Alternativas: Fragmentación Horizontal

J

J₁: Proyectos con presupuesto menor que \$200,000.				
J₂: Proyectos con presupuesto mayor o igual a \$200,000.				
JNO	JNAME	BUDGET	LOC	
J1	Instrumentation	150000	Montreal	
J2	Database Develop.	135000	New York	

J₁

JNO	JNAME	BUDGET	LOC	
J1	Instrumentation	150000	Montreal	
J2	Database Develop.	135000	New York	

J₂

JNO	JNAME	BUDGET	LOC	
J3	CAD/CAM	250000	New York	
J4	Maintenance	310000	Paris	
J5	CAD/CAM	500000	Boston	

Alternativas: Fragmentación Vertical

J₁: Información de los presupuestos de los proyectos.

J₂: Información de los nombres de los proyectos y su ubicación.

J				
JNO	JNAME	BUDGET	LOC	
J1	Instrumentation	150000	Montreal	
J2	Database Develop.	135000	New York	
J3	CAD/CAM	250000	New York	
J4	Maintenance	310000	Paris	
J5	CAD/CAM	500000	Boston	

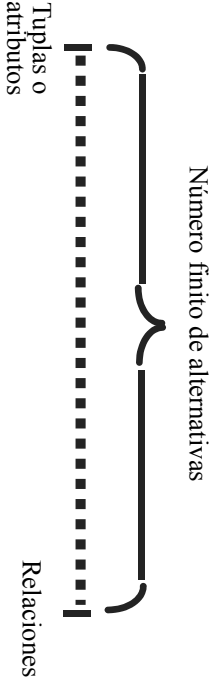
J₁

JNO	BUDGET
J1	150000
J2	135000
J3	250000
J4	310000
J5	500000

J₂

JNO	JNAME	LOC
J1	Instrumentation	Montreal
J2	Database Develop.	New York
J3	CAD/CAM	New York
J4	Maintenance	Paris
J5	CAD/CAM	Boston

Grado de Fragmentación



Encontrar el nivel adecuado de particionamiento dentro de este rango.

Correctez de la Fragmentación

- Completez.
 - ◆ La descomposición de una relación R en fragmentos $R_1, R_2, \dots, R_n$, es completa si y sólo si cada elemento de datos en R puede ser encontrado en algún R_i .
- Reconstrucción.
 - ◆ Si la relación R se descompone en fragmentos $R_1, R_2, \dots, R_n$, debiera existir un operador relacional ∇ tal que $R = \nabla R_i, R_i \in F_R$.
- Excluyente.
 - ◆ Si la relación R se descompone en fragmentos $R_1, R_2, \dots, R_n$, y datos del elemento d_i están en R_j , entonces d_i no debiera estar en algún otro fragmento R_k ($i \neq k$).

M.C. Víctor J. Sosa Sosa

Alternativas de Asignación

- No Replicado.
 - ◆ Particionada: cada fragmento reside en un solo sitio.
- Replicado.
 - ◆ Réplica completa: cada fragmento en cada sitio.
 - ◆ Réplica parcial: cada fragmento en algunos de los sitios.
- Regla del pulgar:
 - Si $\frac{\text{Consultas sólo de lectura}}{\text{Consultas para actualizar}} \geq 1$ replicación es ventajosa, de lo contrario la réplica puede causar problemas.

M.C. Víctor J. Sosa Sosa

Comparación de Alternativas de Réplica

	Réplica Total	Réplica Parcial	Particionado
PROCESAMIENTO DE CONSULTAS	Fácil	Misma ↔	Dificultad →
ADMINISTRACION DE DICCIONARIO	Fácil o no existente	Misma ↔	Dificultad →
CONTROL DE CONCURRENCIA	Moderado	Difícil	Fácil
FIABILIDAD	Muy Alta	Alta	Baja
FACTIBILIDAD	Posible Aplicación	Realista	Posible Aplicación

M.C. Víctor J. Sosa Sosa

Requerimientos de Información

- ☐ Cuatro Categorías:
- ◆ Información de la base de datos.
 - ◆ Información de la aplicación.
 - ◆ Información de la red de comunicaciones.
 - ◆ Información del sistema de cómputo.

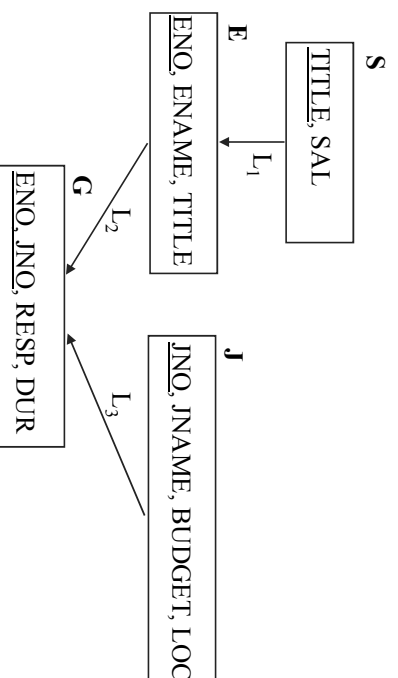
M.C. Víctor J. Sosa Sosa

Tipos de Fragmentación

- Fragmentación horizontal (FH).
- ◆ Fragmentación horizontal primaria (FHP).
- ◆ Fragmentación horizontal derivada (FHD).
- Fragmentación vertical (FV).
- Fragmentación híbrida o mixta (FM).

FHP - Requerimientos de Información

- Información de la base de datos:
- ◆ Relaciones



- ◆ Cardinalidad de cada relación: $card(R)$.

FHP - Requerimientos de Información

□ Información de la aplicación:

- ♦ Predicados simples: Dada $R[A_1, A_2, \dots, A_n]$, un predicado simple p_j es

$$p_j : A_i \theta Valor$$

donde $\theta \in \{=, \neq, <, \leq, >, \geq\}$, $Valor \in D_i$ y D_i es el dominio de A_i .

Ejemplo: JNAME = 'Maintenance'

BUDGET $\leq$ 200000.

- ♦ Predicados minterm: Dado R_i y $Pr_i = \{p_{i1}, p_{i2}, \dots, p_{im}\}$ definir $M_i = \{m_{ij} \mid m_{ij} = \mathbf{V}_{p_{ik} \in Pr_i} p_{ik}^*\}$, $1 \leq k \leq m$, $1 \leq j \leq z$

donde $p_{ik}^* = p_i \circ p_{ik}^* = \neg p_i$.

M.C. Victor J. Sosa Sosa

FHP - Requerimientos de Información

Ejemplos de predicados minterm:

$m_1 : \text{JNAME} = \text{'Maintenance'} \wedge \text{BUDGET} \leq 200000$

$m_2 : \neg(\text{JNAME} = \text{'Maintenance'}) \wedge \text{BUDGET} \leq 200000$

$m_3 : \text{JNAME} = \text{'Maintenance'} \wedge \neg(\text{BUDGET} \leq 200000)$

$m_4 : \neg(\text{JNAME} = \text{'Maintenance'}) \wedge \neg(\text{BUDGET} \leq 200000)$

M.C. Victor J. Sosa Sosa

FHP - Requerimientos de Información

- Información de la aplicación:
 - ♦ Selectividad de predicados minterm: $sel(m_j)$
Número de tuplas de la relación que son accedidas por una consulta de usuario la cual está especificada por un predicado minterm m_j dado.
 - ♦ Frecuencias de acceso: $acc(q_j)$
Frecuencia con la cual la aplicación q_j accesa datos.
Análogamente se puede definir la frecuencia de acceso para un predicado minterm.

M.C. Víctor J. Sosa Sosa

Fragmentación Horizontal Primaria

Definición:

$$R_{ij} = \sigma_{F_j}(R_j), \quad 1 \leq j \leq w$$

donde F_j es la fórmula de selección usada para obtener el fragmento R_{ij} ; el cual es (preferentemente) un predicado minterm.

Por lo tanto:

- ♦ Una fragmentación horizontal R_j de una relación R consiste de todas las tuplas de R las cuales satisfacen un predicado minterm.
- ♦ Dado un conjunto de predicados minterm M , hay tantas fragmentaciones horizontales de la relación R como predicados minterm existan.
- ♦ A los fragmentos horizontales también se les llama fragmentos minterm.

M.C. Víctor J. Sosa Sosa

FHP - Obtención de Predicados Simples

Dados: Una relación R , y el conjunto de predicados simples Pr .

Obtener: El conjunto de fragmentos de $R = \{R_j, R_2, \dots, R_w\}$ los cuales obedecen las reglas de fragmentación.

Preliminares:

- ♦ Pr debe ser completo.
- ♦ Pr debe ser mínimo.

M.C. Víctor J. Sosa Sosa

Completez de los Predicados Simples

Se dice que un conjunto de predicados simples Pr es completo si y sólo si el acceso a las tuplas de los fragmentos minterm definidos sobre Pr requiere que dos tuplas del mismo fragmento minterm tengan la misma probabilidad de ser accesadas por todas y cada una de las aplicaciones.

M.C. Víctor J. Sosa Sosa

Completez de los Predicados Simples

Ejemplo:

Suponemos que $J(JNO, JNAME, BUDGET, LOC)$ tiene dos aplicaciones definidas para trabajar sobre J :

- ➊ Encontrar los presupuestos de los proyectos en cada localidad.
- ➋ Encontrar los proyectos con presupuesto menor que \$200000.

M.C. Victor J. Sosa Sosa

Completez de los Predicados Simples

De acuerdo a (1)

$P_r = \{LOC = \text{“Montreal”}, LOC = \text{“New York”}, LOC = \text{“Paris”}\}$

lo cual no es completo con respecto a (2).

Si modificamos

$P_r = \{LOC = \text{“Montreal”}, LOC = \text{“New York”}, LOC = \text{“Paris”}, BUDGET \leq 200000, BUDGET > 200000\}$

el cual es completo.

M.C. Victor J. Sosa Sosa

Minimalidad de los Predicados Simples

- Si un predicado influye en cómo se ejecuta la fragmentación (v.g. provoca que un fragmento f se particione en nuevos fragmentos f_i y f_j), entonces debiera existir al menos una aplicación que accese a f_i y f_j diferentemente.
- En otras palabras el predicado simple deberá ser *relevante* en determinar una fragmentación.
- Si todos los predicados de un conjunto Pr son relevantes, entonces Pr es mínimo.

M.C. Víctor J. Sosa Sosa

Relevancia de los Predicados Simples

Sean m_i y m_j dos predicados mintern idénticos en su definición, excepto que m_i contiene el predicado simple p_i mientras que m_j contiene $\neg p_i$, y sean f_i y f_j dos fragmentos definidos de acuerdo a m_i y m_j ; entonces p_i es relevante si y sólo si

$$\frac{acc(m_i)}{card(f_i)} \neq \frac{acc(m_j)}{card(f_j)}$$

M.C. Víctor J. Sosa Sosa

Minimalidad de los Predicados Simples

Ejemplo

$Pr = \{ \text{LOC} = \text{"Montreal"}, \text{LOC} = \text{"New York"}, \text{LOC} = \text{"Paris"}, \text{BUDGET} \leq 200000, \text{BUDGET} > 200000 \}$
es mínimo además de ser completo.

Sin embargo, si agregamos

$\text{JNAME} = \text{"Instrumentation"}$

entonces Pr no es mínimo.

M.C. Victor J. Sosa Sosa

Algoritmo COM-MIN

Entrada: Una relación R y un conjunto de predicados simples Pr .

Salida: Un conjunto de predicados simples completo y mínimo Pr' para Pr .

Regla I: Una relación o fragmento se particiona en al menos dos partes las cuales son accedadas de manera diferente por al menos una aplicación.

M.C. Victor J. Sosa Sosa

Algoritmo COM-MIN

- ❶ Inicialización:
 - ◆ Encontrar un $p_i \in Pr$ tal que p_i particione R de acuerdo a la Regla 1.
 - ◆ Hacer $Pr' \leftarrow p_i$; $Pr \leftarrow Pr - p_i$; $F \leftarrow f_i$
- ❷ Agregar predicados iterativamente a Pr' hasta que esté completo.
 - ◆ Encontrar un $p_j \in Pr$ tal que p_j particione algún f_k definido por un predicado minterm de Pr' de acuerdo a la Regla 1.
 - ◆ Hacer $Pr' \leftarrow Pr' \cup p_j$; $Pr \leftarrow Pr - p_j$; $F \leftarrow F \cup f_i$
 - ◆ Si $\exists p_k \in Pr'$ el cual no es relevante, entonces Hacer $Pr' \leftarrow Pr' - p_k$; $F \leftarrow F - f_k$

M.C. Victor J. Sosa Sosa

Algoritmo P-HORIZONTAL

Hacer uso de COM-MIN para efectuar fragmentación.

Entrada: Una relación R y un conjunto de predicados simples Pr .

Salida: Un conjunto de fragmentos minterm M de R .

- ❶ Hacer $Pr' \leftarrow \text{COM-MIN}(R, Pr)$
- ❷ Determinar el conjunto M de predicados minterm
- ❸ Determinar el conjunto de implicaciones I entre $p_i \in Pr$
- ❹ Eliminar los predicados minterm contradictorios de M

M.C. Victor J. Sosa Sosa

FHP - Ejemplo

- Dos relaciones candidatas: S y J.
- Fragmentación de la relación S:
 - ♦ Aplicación: verificar salario y determinar aumento.
 - ♦ Los registros de empleados se manipulan en dos sitios => la aplicación corre en dos sitios.
 - ♦ Predicados simples:
 $p_1 : SAL \leq 30000$
 $p_2 : SAL > 30000$
 $Pr : \{p_1, p_2\}$ el cual es completo y mínimo $Pr' = Pr$
 - ♦ Predicados minterm:
 $m_1 : (SAL \leq 30000) \wedge (SAL > 30000)$
 $m_2 : (SAL \leq 30000) \wedge \neg (SAL > 30000)$
 $m_3 : \neg (SAL \leq 30000) \wedge (SAL > 30000)$
 $m_4 : \neg (SAL \leq 30000) \wedge \neg (SAL > 30000)$

FHP - Ejemplo

- Fragmentación de la relación S (continúa):
 - ♦ Implicaciones:
 $i_1 : (SAL \leq 30000) \Rightarrow \neg (SAL > 30000)$
 $i_2 : \neg (SAL \leq 30000) \Rightarrow (SAL > 30000)$
 $i_3 : (SAL > 30000) \Rightarrow \neg (SAL \leq 30000)$
 $i_4 : \neg (SAL > 30000) \Rightarrow (SAL \leq 30000)$
 - ♦ m_1 es contradictorio a i_1 , m_4 es contradictorio a i_2

S₁

TITLE	SAL
Mech. Eng. Programmer	27000 24000

S₂

TITLE	SAL
Elect. Eng. Syst. Anal.	40000 34000

FHP - Ejemplo

- Fragmentación de la relación J:
 - ♦ Aplicaciones:
 - ➊ Encontrar el nombre y el presupuesto de los proyectos dado su número. Estas consultas son emitidas en tres sitios.
 - ➋ Accesar la información del proyecto de acuerdo a su presupuesto.
 - ♦ Predicados simples para la aplicación (1):
 - p_1 : LOC = "Montreal"
 - p_2 : LOC = "New York"
 - p_3 : LOC = "Paris"
 - ♦ Predicados simples para la aplicación (2):
 - p_4 : BUDGET $\leq$ 200000
 - p_5 : BUDGET $>$ 200000
 - ♦ $Pr = Pr' = \{p_1, p_2, p_3, p_4, p_5\}$

M.C. Víctor J. Sosa Sosa

FHP - Ejemplo

- Fragmentación de la relación J (continúa):
 - ♦ Fragmentos mintern después de la eliminación:
 - m_1 : (LOC = "Montreal") $\wedge$ (BUDGET $\leq$ 200000)
 - m_2 : (LOC = "Montreal") $\wedge$ (BUDGET $>$ 200000)
 - m_3 : (LOC = "New York") $\wedge$ (BUDGET $\leq$ 200000)
 - m_4 : (LOC = "New York") $\wedge$ (BUDGET $>$ 200000)
 - m_5 : (LOC = "Paris") $\wedge$ (BUDGET $\leq$ 200000)
 - m_6 : (LOC = "Paris") $\wedge$ (BUDGET $>$ 200000)

M.C. Víctor J. Sosa Sosa

FHP - Ejemplo

J₁

JNO	JNAME	BUDGET	LOC
J1	Instrumentation	150000	Montreal

J₂

JNO	JNAME	BUDGET	LOC
J2	Database Develop.	135000	New York

J₄

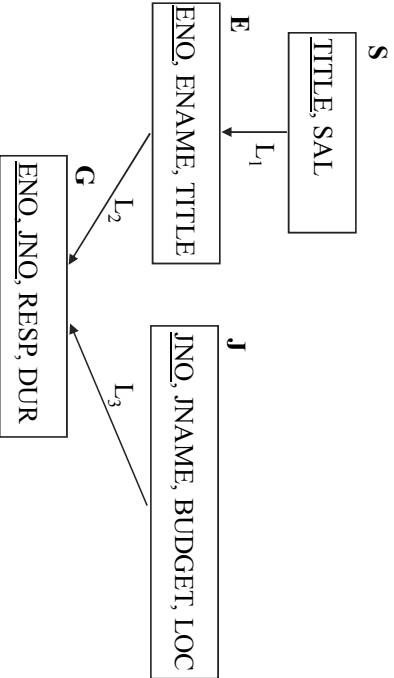
JNO	JNAME	BUDGET	LOC
J3	CAD/CAM	250000	New York

J₆

JNO	JNAME	BUDGET	LOC
J4	Maintenance	310000	Paris

Fragmentación Horizontal Derivada

- ❑ Definida sobre una relación miembro de una liga según una operación de selección especificada sobre su relación propietaria.
- ♦ Cada enlace corresponde a una equireunión.



FHD - Definición

Dado un enlace L donde *propietario*(L)=S y *miembro*(L)=R, la fragmentación horizontal derivada de R se define como

$$R_i = R \bowtie S_p \quad 1 \leq i \leq w$$

Donde w es el número máximo de fragmentos que serán definidos en R, y $S_i = \sigma_{F_i}(S)$, donde F_i es la formula mediante la cual se define la fragmentación horizontal primaria S_i .

FHD - Ejemplo

Dado el enlace L_I , donde *propietario*(L_I)=S y *miembro*(L_I)=E

$$\begin{aligned} E_1 &= E \bowtie S_1 \\ E_2 &= E \bowtie S_2 \end{aligned}$$

donde

$$\begin{aligned} S_1 &= \sigma_{SAL \leq 30000}(S) \\ S_2 &= \sigma_{SAL > 30000}(S) \end{aligned}$$

E_1

	ENO	ENAME	TITLE
E3	A. Lee	Mech. Eng.	Programmer
E4	J. Miller		
E7	R. Davis	Mech. Eng.	

E_2

	ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng.	Syst. Anal.
E2	M. Smith	Syst. Anal.	
E5	B. Casey	Syst. Anal.	
E6	L. Chu	Elect. Eng.	
E8	J. Jones	Syst. Anal.	

FHD - Correctez

- Completez.
 - ♦ Integridad referencial.
 - ♦ Sea R la relación miembro de un enlace cuyo propietario es la relación S la cual está fragmentada como $F_s = \{S_l, S_2, \dots, S_w\}$. Además sea A el atributo de la reunión entre R y S . Entonces para cada tupla t de R , debiera existir una tupla t' de S tal que $t[A] = t'[A]$
- Reconstrucción.
 - ♦ Igual que la de la fragmentación horizontal primaria.
- Exclusión.
 - ♦ Si el grafo de la reunión entre el propietario y los fragmentos miembro es simple.

M.C. Victor J. Sosa Sosa

Fragmentación Vertical

- Ha sido estudiada dentro del contexto centralizado.
 - ♦ Metodología de diseño.
 - ♦ Agrupamiento físico.
- Más difícil que la horizontal, debido a que existen más alternativas.
- Existen dos técnicas:
 - ♦ Agrupamiento de atributos en fragmentos.
 - ♦ División de la relación en fragmentos.

M.C. Victor J. Sosa Sosa

Fragmentación Vertical

- Traslape de fragmentos.
 - ◆ Agrupamiento.
- Sin traslape de fragmentos.
 - ◆ División.

No se considera traslape en los atributos llave replicados.

Ventaja: facilita hacer cumplir las dependencias funcionales (para verificar integridad, etc.)

M.C. Víctor J. Sosa Sosa

FV - Requerimientos de Información

- Información de la aplicación:
 - ◆ Afinidad de los atributos, la cual es una medida que indica qué tanta relación existe entre los atributos.
 - ◆ Esto se obtiene de las primitivas de uso de los datos.
- Dado un conjunto de consultas $Q = \{q_1, q_2, \dots, q_q\}$ que corren sobre la relación $R[A_1, A_2, \dots, A_n]$,

$$uso(q_i, A_j) = \begin{cases} 1 & \text{si el atributo } A_j \text{ es referido en la consulta } q_i \\ 0 & \text{en caso contrario} \end{cases}$$

$uso(q_i, \bullet)$ se puede definir análogamente.

M.C. Víctor J. Sosa Sosa

FV - Ejemplo de *uso*(q_i, A_j)

Considerar las siguientes 4 consultas para la relación J:

q_1 :	SELECT BUDGET	q_2 :	SELECT JNAME, BUDGET
	FROM J		FROM J
	WHERE JNO=Valor		
q_3 :	SELECT JNAME	q_4 :	SELECT SUM(BUDGET)
	FROM J		FROM J
	WHERE LOC=Valor		WHERE LOC=Valor

Sean A_1 =JNO, A_2 =JNAME, A_3 =BUDGET, A_4 =LOC

$$\begin{matrix} & A_1 & A_2 & A_3 & A_4 \\ \begin{matrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{matrix} & \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \end{matrix}$$

FV - Medida de Afinidad $aff(A_i, A_j)$

La medida de afinidad entre dos atributos A_i y A_j de una relación $R[A_1, A_2, \dots, A_n]$ con respecto al conjunto de aplicaciones $\mathcal{Q} = \{q_1, q_2, \dots, q_n\}$ se define como sigue

$$aff(A_i, A_j) = \sum_{\text{todas las consultas que accedan } A_i \text{ y } A_j} \sum_{v \in S_i} ref(q_k) acc(q_k)$$

FV - Cálculo de $aff(A_i, A_j)$

Suponemos que cada consulta en el ejemplo anterior accesa los atributos una vez durante cada ejecución.

También suponemos las frecuencias de acceso $\xrightarrow{\hspace{1cm}}$

	S ₁	S ₂	S ₃
q ₁	15	20	10
q ₂	5	0	0
q ₃	25	25	25
q ₄	3	0	0

Entonces

$$aff(A_i, A_j) = 15*1 + 20*1 + 10*1 = 45$$

y la matriz de afinidad de atributos AA es $\xrightarrow{\hspace{1cm}}$

	A ₁	A ₂	A ₃	A ₄
A ₁	45	0	45	0
A ₂	0	80	5	75
A ₃	45	5	53	3
A ₄	0	75	3	78

FV - Algoritmo de Agrupamiento

- Toma la matriz de afinidad de atributos AA y reorganiza el orden de los atributos para formar grupos donde los atributos en cada grupo demuestran alta afinidad unos con otros.
- El algoritmo de energía de cohesión (BEA) se usado para agrupamiento de entidades. BEA encuentra un orden de atributos tal que la medida de afinidad global

$$AM = \sum_i \sum_j (afinidad\ de\ A_i\ y\ A_j\ con\ sus\ vecinos)$$

se maximiza.

Algoritmo de Energía de Cohesión

Entrada: La matriz AA .

Salida: La matriz de afinidad CA agrupada, la cual es una derivación de AA .

- ➊ *Inicialización*: Colocar y ajustar una de las columnas de AA en CA .
- ➋ Iteración: Colocar las $n-i$ columnas restantes en el resto de $i+1$ posiciones en la matriz CA . Por cada columna, elegir el lugar que más contribuya a la medida de afinidad global.
- ➌ Ordenar filas: Ordenar las filas de acuerdo al orden de columna.

M.C. Víctor J. Sosa Sosa

Algoritmo de Energía de Cohesión

La “mayor contribución” se encuentra de acuerdo a la siguiente fórmula:

$$cont(A_i, A_k, A_j) = 2bond(A_i, A_k) + 2bond(A_k, A_j) - 2bond(A_i, A_j)$$

donde:

$$bond(A_x, A_y) = \sum_{z=1}^n aff(A_z, A_x) aff(A_z, A_y)$$

M.C. Víctor J. Sosa Sosa

BEA - Ejemplo

Considerar la siguiente matriz AA y la correspondiente matriz CA donde A_1 y A_2 han sido ubicados como sigue:

$$AA = \begin{matrix} & \begin{matrix} A_1 & A_2 & A_3 & A_4 \end{matrix} \\ \begin{matrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{matrix} & \begin{bmatrix} 45 & 0 & 5 & 0 \\ 0 & 80 & 5 & 75 \\ 45 & 5 & 53 & 3 \\ 0 & 75 & 3 & 78 \end{bmatrix} \end{matrix} \qquad CA = \begin{matrix} & \begin{matrix} A_1 & A_2 \end{matrix} \\ \begin{matrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{matrix} & \begin{bmatrix} 45 & 0 \\ 0 & 80 \\ 45 & 5 \\ 0 & 75 \end{bmatrix} \end{matrix}$$

Ordenar (0-3-1):

$$cont(A_0, A_3, A_1) = 2bond(A_0, A_3) + 2bond(A_3, A_1) - 2bond(A_0, A_1) \\ = 2*0 + 2*4410 - 2*0 = 8820$$

Ordenar (1-3-2):

$$cont(A_1, A_3, A_2) = 2bond(A_1, A_3) + 2bond(A_3, A_2) - 2bond(A_1, A_2) \\ = 2*4410 + 2*890 - 2*225 = 10150$$

Ordenar (2-3-4):

$$cont(A_2, A_3, A_4) = 1780$$

BEA - Ejemplo

Por lo tanto la matriz CA tiene la forma

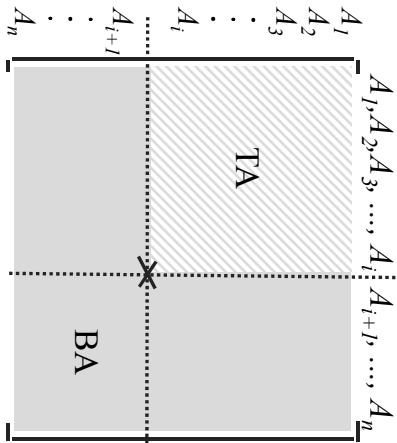
$$\begin{matrix} & \begin{matrix} A_1 & A_3 & A_2 & A_4 \end{matrix} \\ \begin{matrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{matrix} & \begin{bmatrix} 45 & 5 & 0 & \\ 0 & 5 & 80 & \\ 45 & 53 & 5 & \\ 0 & 3 & 75 & \end{bmatrix} \end{matrix}$$

La forma final de la matriz CA (después de organizar las filas) es

$$\begin{matrix} & \begin{matrix} A_1 & A_3 & A_2 & A_4 \end{matrix} \\ \begin{matrix} A_1 \\ A_3 \\ A_2 \\ A_4 \end{matrix} & \begin{bmatrix} 45 & 45 & 0 & 0 \\ 45 & 53 & 5 & 3 \\ 0 & 5 & 80 & 75 \\ 0 & 3 & 75 & 78 \end{bmatrix} \end{matrix}$$

FV - Algoritmo

¿Cómo puede dividirse un conjunto de atributos agrupados $\{A_1, A_2, \dots, A_n\}$ en dos (o más) conjuntos $\{A_1, A_2, \dots, A_i\}$ y $\{A_{i+1}, \dots, A_n\}$ tal que no haya (o sea mínimo el número de) aplicaciones que accesen ambos (o más de un) conjuntos?



M.C. Víctor J. Sosa Sosa

FV - Algoritmo

Definir

TQ = conjunto de las aplicaciones que accesan sólo TA ,
 BQ = conjunto de las aplicaciones que accesan sólo BA ,
 OQ = conjunto de las aplicaciones que accesan TA y BA ,
y

CTQ = número total de accesos a los atributos por las aplicaciones que accesan sólo TA ,
 CBQ = número total de accesos a los atributos por las aplicaciones que accesan sólo BA ,
 COQ = número total de accesos a los atributos por las aplicaciones que accesan TA y BA .

Entonces encontrar el punto sobre la diagonal que maximiza

$$CTQ*CBQ-COQ^2$$

M.C. Víctor J. Sosa Sosa

FV - Algoritmo

Dos problemas:

- Un fragmento a la mitad de la matriz CA .
 - ♦ Cambiar la columna más izquierda hasta la derecha, reordenar las filas, y aplicar el algoritmo para encontrar el “mejor” punto de particionamiento.
 - ♦ Hacer esto para todos los cambios posibles.
 - ♦ Costo $O(n^2)$.
- Para más de dos grupos.
 - ♦ Particionamiento en n grupos.
 - ♦ Probar $1, 2, \dots, n-1$ puntos de división a lo largo de la diagonal y tratar de encontrar el mejor punto para cada uno de éstos.
 - ♦ Costo $O(2^n)$.

M.C. Víctor J. Sosa Sosa

FV - Correctez

Una relación R , definida sobre un conjunto de atributos A y clave K , genera la partición vertical $F_R = \{R_1, R_2, \dots, R_r\}$.

- Completez. Lo siguiente debiera satisfacerse:

$$A = \cup A_i$$

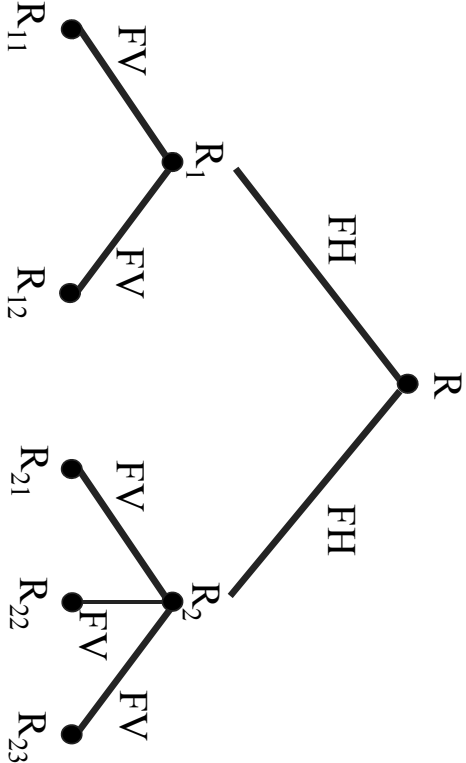
- Reconstrucción. La reconstrucción puede efectuarse mediante

$$R = \bowtie_K R_i, \forall R_i \in F_R$$

- Exclusión.
 - ♦ Los identificadores no se consideran para ser trasladados ya que son mantenidos por el sistema.
 - ♦ Las llaves duplicadas no se consideran para ser trasladadas.

M.C. Víctor J. Sosa Sosa

Fragmentación Híbrida



Distribución de Fragmentos

Planteamiento del problema:

Dado

$F = \{F_1, F_2, \dots, F_n\}$ es el conjunto de fragmentos.
 $S = \{S_1, S_2, \dots, S_m\}$ es el conjunto de sitios en la red, y
 $Q = \{q_1, q_2, \dots, q_q\}$ es el conjunto de aplicaciones,
encontrar la distribución “óptima” de F en S.

Distribución de Fragmentos

Optimalidad

- Costo mínimo.
 - ◆ Comunicación + Almacenamiento + Procesamiento (lectura y actualizaciones).
 - ◆ Costo en términos de tiempo (usualmente).
- Eficiencia.
 - ◆ Tiempo de respuesta.
- Restricciones.
 - ◆ Restricciones por sitio (almacenaje y procesamiento).

M.C. Víctor J. Sosa Sosa

Requerimientos de Información

- Información de la base de datos.
 - ◆ Selectividad de fragmentos.
 - ◆ Tamaño de un fragmento.
- Información de la aplicación.
 - ◆ Tipos de acceso y número.
 - ◆ Acceso a sitios.
- Información de la red de comunicaciones.
 - ◆ Ancho de banda.
 - ◆ Distancias entre sitios.
 - ◆ Sobrecarga de protocolos.
- Información del sistema de cómputo.
 - ◆ Unidad de costo de almacenar datos en un sitio.
 - ◆ Unidad de costo de procesar datos en un sitio.

M.C. Víctor J. Sosa Sosa

Distribución

Problema de distribución de archivos (FAP) vs. problema de distribución de base de datos (DAP).

- Los fragmentos no pueden tratarse como archivos individuales.
 - ◆ Se tienen que tomar en cuenta sus relaciones.
- El acceso a la base de datos es más complicado.
 - ◆ El modelo de acceso a archivos remotos no aplica.
 - ◆ Relación entre distribución y procesamiento de consultas.
- El costo de mantener la integridad debe considerarse.
- El costo del control de concurrencia debe considerarse.

M.C. Víctor J. Sosa Sosa

Distribución - Requerimientos de Información

- Información de la base de datos:
 - ◆ Selectividad de fragmentos minterm.
 - ◆ Tamaño de los fragmentos.
- Información de las aplicaciones:
 - ◆ Número de accesos de lectura de una consulta a un fragmento.
 - ◆ Número de accesos de actualización a un fragmento.
 - ◆ Una matriz que indique cuáles consultas actualizan qué fragmentos.
 - ◆ Una matriz similar para las lecturas.
 - ◆ Sitio de origen de cada consulta.

M.C. Víctor J. Sosa Sosa

Distribución - Requerimientos de Información

- Información de los sitios:
 - ♦ Unidad de costo de almacenar datos en un sitio.
 - ♦ Unidad de costo de procesar datos en un sitio.
- Información de la red:
 - ♦ Costo de comunicación de una trama entre dos sitios.
 - ♦ Tamaño de la trama.

M.C. Víctor J. Sosa Sosa

Modelo de Distribución

Forma general

min(costo total)

sujeto a

restricciones de tiempo de respuesta

restricción de almacenaje

restricción de procesamiento

variables de decisión:

$$x_{ij} = \begin{cases} 1 & \text{si el fragmento } F_i \text{ se almacena en el sitio } S_j \\ 0 & \text{en caso contrario} \end{cases}$$

M.C. Víctor J. Sosa Sosa

Modelo de Distribución

- Costo Total :
 $\sum_{\text{todas las consultas}} \text{costo de procesamiento de consulta} + \sum_{\text{todos los sitios}} \sum_{\text{todos los fragmentos}} \text{costo de almacenaje de un frag. en un sitio}$
- Costo de almacenaje (del fragmento F_j en S_k):
(unidad de costo de almacenaje en S_k) * (tamaño de F_j) * x_{jk}
- Costo de procesamiento de consulta:
componente de procesamiento + componente de transmisión

M.C. Víctor J. Sosa Sosa

Modelo de Distribución

- Componente de procesamiento:
costo de acceso + costo de mantenimiento de integridad
+ costo de control de concurrencia
- Costo de Acceso:
 $\sum_{\text{todos los sitios}} \sum_{\text{todos los fragmentos}} (\text{no. de accesos de actualización} + \text{no. de accesos de lectura}) * x_{jk}$
* costo de procesamiento local en un sitio
- Los costos de mantenimiento de integridad y de control de concurrencia se pueden formular de una manera similar.

M.C. Víctor J. Sosa Sosa

Modelo de Distribución

- Componente de Transmisión:
costo de procesar actualizaciones + costo de procesar lecturas
- Costo de actualizaciones:
 $\sum_{\text{todos los sitios}} \sum_{\text{todos los fragmentos}} \text{costo del mensaje de actualizar} + \sum_{\text{todos los sitios}} \sum_{\text{todos los fragmentos}} \text{costo de confirmación}$
- Costo de lecturas:
 $\sum_{\text{todos los fragmentos}} \min_{\text{todos los sitios}} (\text{costo del comando de recuperación} + \text{costo de enviar los resultados})$

M.C. Víctor J. Sosa Sosa

Modelo de Distribución

- Restricciones:
 - ♦ Tiempo de respuesta:
tiempo de ejecución de la consulta
 $\leq \text{max. tiempo de resp. permisible para esa consulta}$
 - ♦ Almacenamiento:
 $\sum_{\text{todos los fragmentos}} \text{reqs. de almacenaje de un fragmento en ese sitio} \leq \text{capacidad de almacenaje de ese sitio}$
 - ♦ Procesamiento:
 $\sum_{\text{todas las consultas}} \text{carga de proc. de una consultas en ese sitio} \leq \text{capacidad de procesamiento de ese sitio}$

M.C. Víctor J. Sosa Sosa

Problema - Separación de los Pasos de Diseño

