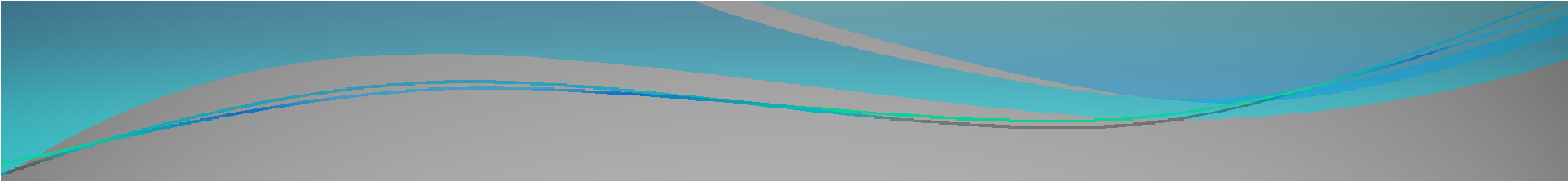




Diseño de Bases de Datos Distribuidas

- 
- 1. El proceso de diseño de una base de datos distribuida**
 - 2. Objetivos del diseño de la distribución de datos**
 - 3. Estrategias de diseño de una base de datos distribuida**
 - 4. Tipos de fragmentación de datos**
 - 5. Diseño de la fragmentación de una base de datos**
 - 6. Criterios generales para la asignación de fragmentos**
 - 7. Medida del coste y beneficio de la asignación de fragmentos**

El proceso de diseño de una base de datos distribuida

El diseño de una base de datos centralizada consiste en

- Diseño del esquema conceptual
- Diseño del esquema lógico
- Diseño del esquema físico

En una base de datos distribuida, estas etapas equivalen al

- Diseño del esquema conceptual
- Diseño del esquema lógico global
- Diseño de los esquemas físicos locales

La distribución de una base de datos añade:

- Diseño de la fragmentación  “criterio lógico” que motiva la fragmentación
- Diseño de la asignación de fragmentos  localización “física” de los datos

El proceso de diseño de una base de datos distribuida

Requerimientos de las aplicaciones más "importantes"

- Localidad de origen de la aplicación
- Frecuencia de activación de cada aplicación en cada localidad
- Número, tipo y distribución estadística de los accesos de cada aplicación a cada objeto de datos

Objetivos del diseño de la distribución de los datos

Procesamiento local

Distribuir los datos de forma que se maximice el procesamiento local



Considerar dos tipos de referencias a datos

Referencias locales y referencias remotas

Maximizar el procesamiento local



Minimizar el número de referencias remotas

Aplicación **completamente local**:

Aplicación que puede ejecutarse completamente en su localidad de origen

Objetivos del diseño de la distribución de los datos

■ Disponibilidad y fiabilidad de los datos distribuidos

Alto grado de disponibilidad (para aplicaciones de sólo lectura) y fiabilidad puede conseguirse almacenando múltiples copias de los mismos datos

■ Distribución de la carga de trabajo

Aprovechar la potencia y uso de los computadores de cada localidad, y maximizar el grado de paralelismo en la ejecución de las aplicaciones



Puede afectar negativamente al procesamiento local

■ Costes y disponibilidad de almacenamiento

No es un factor relevante, pero debería tenerse en cuenta

estrategias de diseño de una base de datos distribuida

■ Estrategia descendente (top-down)



Usada para el diseño de

Bases de datos distribuidas autonómicas

■ Estrategia ascendente (bottom-up)

Basada en la "integración" de esquemas existentes



- Combinación de definiciones de datos comunes
- Resolución de conflictos entre diferentes representaciones de los mismos datos



Usada para el diseño de

Bases de datos federadas

estrategias de diseño de una base de datos distribuida

Estrategia descendente (top-down)

Más abstracción




Menos abstracción

- Diseño del esquema conceptual
- Diseño del esquema lógico
- Optimización (Normalización)
- Diseño de la fragmentación
- Diseño de la asignación de fragmentos
- Diseño de los esquemas físicos locales

estrategias de diseño de una base de datos distribuida

Estrategia ascendente (bottom-up)

Más abstracción



Menos abstracción

➤ Integrar esquemas de exportación:
Obtener **esquema federado**

➤ Generar **esquemas de exportación**

➤ Trasladar esquemas locales al MC:
Obtener **esquemas componentes**

➤ Seleccionar modelo de referencia (MC)

➤ Esquemas locales (lógicos)

➤ Esquemas físicos locales

Si sistema homogéneo,
no son necesarias

Tipos de fragmentación de datos

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
<i>a1</i>	<i>b1</i>	<i>c1</i>	<i>d1</i>	<i>e1</i>
<i>a2</i>	<i>b2</i>	<i>c2</i>	<i>d2</i>	<i>e2</i>
...
...
...
...
<i>a_n</i>	<i>b_n</i>	<i>c_n</i>	<i>d_n</i>	<i>e_n</i>

Fragmentación horizontal

Subconjunto de tuplas

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
<i>a1</i>	<i>b1</i>	<i>c1</i>	<i>d1</i>	<i>e1</i>
<i>a2</i>	<i>b2</i>	<i>c2</i>	<i>d2</i>	<i>e2</i>
...
...
...
...
<i>a_n</i>	<i>b_n</i>	<i>c_n</i>	<i>d_n</i>	<i>e_n</i>

Fragmentación vertical

Subconjunto de atributos

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
<i>a1</i>	<i>b1</i>	<i>c1</i>	<i>d1</i>	<i>e1</i>
<i>a2</i>	<i>b2</i>	<i>c2</i>	<i>d2</i>	<i>e2</i>
...
...
...
...
<i>a_n</i>	<i>b_n</i>	<i>c_n</i>	<i>d_n</i>	<i>e_n</i>

Fragmentación mixta

Aplicar fragmentación vertical a una fragmentación horizontal (o viceversa)

Tipos de fragmentación de datos

Fragmento

Expresión en un lenguaje relacional que considera las relaciones globales como operandos y produce como resultado el fragmento

Reglas de fragmentación

◆ Completitud

Todos los datos de la relación global deben transformarse en fragmentos

◆ Reconstrucción

Siempre debe ser posible reconstruir una relación global a partir de sus fragmentos

◆ Fragmentación disjunta

Es conveniente que los fragmentos sean disjuntos

{ Siempre en fragmentos horizontales
No Siempre en fragmentos verticales

Tipos de fragmentación de datos

Fragmentación horizontal

Partir las tuplas de una relación global en subconjuntos

Se define con el operador de **SELECCIÓN** (SL_q)

Se reconstruye con el operador de **UNIÓN** (UN)

A la condición "q" se le denomina **cualificación**

Fragmentación completa

Conjunto de cualificaciones completo con respecto al conjunto de valores permitidos

Fragmentación disjunta

Conjunto de cualificaciones mutuamente excluyente

Ejemplo:

$\text{Proveedor} = (\text{Pnum}, \text{nombre}, \text{ciudad})$

$\text{Proveedor}_1 = SL_{\text{ciudad}='Granada'}(\text{Proveedor})$

$\text{Proveedor}_2 = SL_{\text{ciudad}='Jaén'}(\text{Proveedor})$

$\text{Proveedor}_3 = SL_{\text{ciudad}='Toledo'}(\text{Proveedor})$

$\text{Proveedor} = \text{Proveedor}_1 \text{ UN } \text{Proveedor}_2 \text{ UN } \text{Proveedor}_3$

¿Es esta fragmentación completa y disjunta?

Tipos de fragmentación de datos

Fragmentación horizontal derivada

Fragmentación no basada en los atributos de la relación sino en la fragmentación horizontal de otra relación

Se define con el operador de **SEMI-PRODUCTO NATURAL** (SJN_c)

Se reconstruye con el operador de **UNIÓN** (UN)

Las cualificaciones son condiciones de existencia de tuplas y no incluyen atributos de la relación

Ejemplo:

$Suministro = (Pnum, Anum, Dnum, cantidad)$

$Suministro_1 = Suministro \ SJN_{Pnum=Pnum} \ Proveedor_1$

$Suministro_2 = Suministro \ SJN_{Pnum=Pnum} \ Proveedor_2$

$Suministro_3 = Suministro \ SJN_{Pnum=Pnum} \ Proveedor_3$

$Suministro = Suministro_1 \ UN \ Suministro_2 \ UN \ Suministro_3$

Cualificaciones:

q1: $S.Pnum = P.Pnum$ and $P.ciudad = 'Granada'$

q2: $S.Pnum = P.Pnum$ and $P.ciudad = 'Jaén'$

q3: $S.Pnum = P.Pnum$ and $P.ciudad = 'Toledo'$

Tipos de fragmentación de datos

Fragmentación vertical

Dividir los atributos de una relación global en grupos

Se define con el operador de **POYECCIÓN** (PJ_{Lista})

Se reconstruye con el operador de **PRODUCTO NATURAL** (JN_C)

Para garantizar la reconstrucción se incluye la llave o un identificador de tupla

Fragmentación completa

Si cada atributo se transforma en al menos un atributo en algún fragmento

Ejemplo:

$Empleado = (Enum, nombre, salario, IRPF, mgrnum, Dnum)$

$Empleado_1 = PJ_{Lista1} Empleado$

$Empleado_2 = PJ_{Lista2} Empleado$

$Lista1 = (Enum, nombre, mgrnum, Dnum)$

$Lista2 = (Enum, salario, IRPF)$

$Empleado = Empleado_1 JN_{Enum=Enum} Empleado_2$

¿Es esta fragmentación
disjunta?

Tipos de fragmentación de datos

Fragmentación mixta

Aplicar operaciones de fragmentación recursivamente

Se reconstruye aplicando operaciones de reconstrucción en orden inverso

Ejemplo:

$\text{Empleado} = (\text{Enum}, \text{nombre}, \text{salario}, \text{IRPF}, \text{mgrnum}, \text{Dnum})$

$\text{Empleado}_1 = \text{PJ}_{\text{Lista1}} \text{Empleado}$

$\text{Empleado}_2 = \text{PJ}_{\text{Lista2}} \text{Empleado}$

$\text{Lista1} = (\text{Enum}, \text{nombre}, \text{mgrnum}, \text{Dnum})$

$\text{Lista2} = (\text{Enum}, \text{salario}, \text{IRPF})$

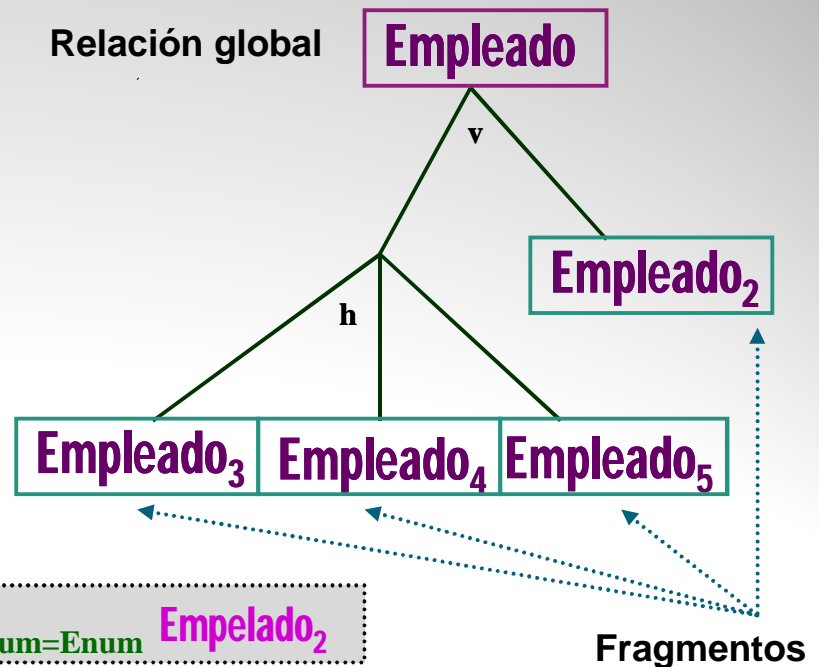
$\text{Empleado}_3 = \text{SL}_{\text{Dnum} \leq 10} (\text{Empleado}_1)$

$\text{Empleado}_4 = \text{SL}_{10 < \text{Dnum} \leq 20} (\text{Empleado}_1)$

$\text{Empleado}_5 = \text{SL}_{\text{Dnum} > 20} (\text{Empleado}_1)$

$\text{Empleado} = (\text{Empleado}_3 \text{ UN } \text{Empleado}_4 \text{ UN } \text{Empleado}_5) \text{ JN}_{\text{Enum}=\text{Enum}} \text{Empleado}_2$

Árbol de fragmentación



Diseño De la fragmentación de una base de datos

Distribución de los datos



Propósito

Determinar los fragmentos que serán
"unidades lógicas de asignación"

Diseño de la fragmentación



Consiste

Agrupar tuplas (o atributos) con las
"mismas propiedades", desde el punto de vista de su asignación



Cada grupo

Fragmento

Ejemplo:

- + *Relación global Empleado*
- + *Aplicación que requiere información sobre los empleados de los departamentos*
- + *Cada departamento es una localidad del Sistema distribuido*
- + *La aplicación se puede generar en cualquier departamento (localidad)*
- + *Cuando se genera en un departamento, la aplicación referencia con mayor probabilidad a los empleados de ese departamento*

Las tuplas de empleados que trabajan en el mismo departamento, tienen las "mismas propiedades", desde el punto de vista de su asignación

Los fragmentos se diseñan agrupando los empleados por departamentos

Diseño De la fragmentación de una base de datos

Fragmentación horizontal

- Determinar propiedades lógicas de los datos
Predicados de fragmentación
- Determinar propiedades estadísticas de los datos
Número de referencias de las aplicaciones a los fragmentos

Fragmentación horizontal primaria

Determinar un conjunto completo y disjunto de predicados de selección

Los elementos que componen cada fragmento deben ser referenciados homogéneamente por las aplicaciones

Diseño De la fragmentación de una base de datos

Para definir la fragmentación horizontal primaria de una relación global, se introducen los siguientes conceptos

■ Predicado simple

Atributo operador de relación **valor**

donde operador de relación = $\{=, \neq, \leq, <, >, \geq\}$

■ Término de predicados

P = conjunto de predicados simples

$$y = \bigwedge_{p_i \in P} p_i^*$$

donde $p_i^* = p_i$ o $\neg p_i$ e $y \neq$ falso

Número de conjunciones:

$$2^n$$

n = número de predicados simples

■ Fragmento

Conjunto de tuplas que verifican un término de predicados

■ Predicado simple relevante

Un predicado simple p_i es relevante con respecto a un conjunto de predicados simples P, si existen al menos dos términos de predicados de P cuyas expresiones difieren, únicamente, en p_i , y tal que, los correspondientes fragmentos son referenciados de manera diferente por al menos una aplicación

Diseño De la fragmentación de una base de datos

Ejemplo:

- ✚ **Relación global Empleado**
- ✚ *Aplicación que requiere información sobre los empleados de los departamentos*
- ✚ *Aplicación que requiere información sobre los empleados que son programadores*
- ✚ *Cada departamento es una localidad del sistema distribuido*
- ✚ *Las aplicaciones se puede generar en cualquier departamento (localidad)*
- ✚ *Por simplicidad se consideran , únicamente dos departamentos*

Conjunto P de predicados simples:

$P = \{Dnum = 1, Job = 'Programador'\}$

(Dnum = 2 es equivalente a Dnum \neq 1)

Términos de predicados:

$y_1 : Dnum = 1 \wedge Job = 'Programador'$

$y_2 : Dnum = 1 \wedge Job \neq 'Programador'$

$y_3 : Dnum \neq 1 \wedge Job = 'Programador'$

$y_4 : Dnum \neq 1 \wedge Job \neq 'Programador'$

Todos los predicados son relevantes

Ejemplo de predicado no relevante

Salario > 5000

¿Por qué?

Diseño De la fragmentación de una base de datos

Propiedades de un conjunto P de predicados simples

■ Completo

Si cualquier par de tuplas contenidas en el mismo fragmento son referenciadas con la misma probabilidad por cualquier aplicación

■ Minimal

Si todos los predicados simples contenidos en él son relevantes

Ejemplo

$P = \{Dnum = 1\}$

Es minimal, pero no completo

$P = \{Dnum = 1, Job = 'Programador'\}$

Es completo y minimal

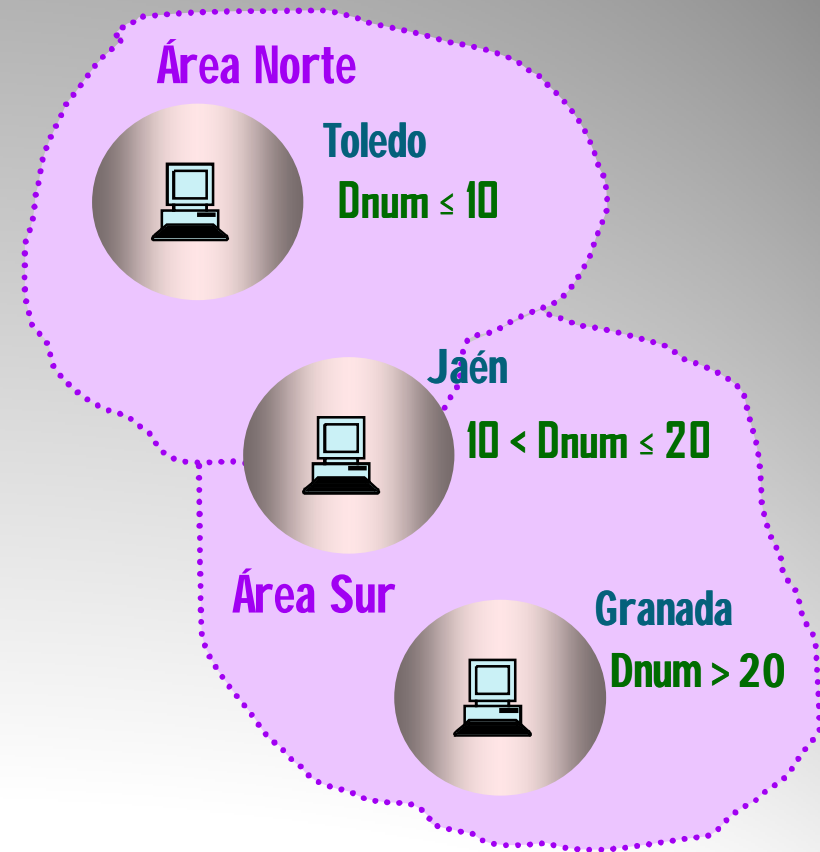
$P = \{Dnum = 1, Job = 'Programador', Salario > 5000\}$

Es completo, pero no minimal

Diseño De la fragmentación de una base de datos

Ejemplo general

- ✚ *Tres localidades: Toledo (localidad 1), Jaén (localidad 2) y Granada (localidad 3)*
- ✚ *30 departamentos: los 10 primeros cerca de Toledo, entre el 11 y el 20 cerca de Jaén y los 10 últimos cerca de Granada*
- ✚ *Los proveedores están todos en Toledo o en Granada*
- ✚ *Dos áreas: el área Norte que incluye Toledo, el área Sur que incluye Granada, y Jaén que se localiza exactamente en la frontera*
- ✚ *Algunos departamentos cercanos a Jaén se encuentran en el área Norte y el resto en el área Sur*



Esquemas globales

Proveedor = (Pnum, nombre, ciudad)

Suministro = (Pnum, Anum, Dnum, cantidad)

Empleado = (Enum, nombre, salario, IRPF, mgrnum, Dnum)

Departamento = (Dnum, nombre, área, mgrnum)

Diseño De la fragmentación de una base de datos

Ejemplo general (continuación)

Relación global a fragmentar

Proveedor = (Pnum, nombre, ciudad)

Aplicación

SELECT nombre FROM Proveedor WHERE Pnum = \$X

- ✚ Si se genera en Toledo o en Granada, se referencian a los proveedores de esas ciudades con una probabilidad del 80%
- ✚ Si se genera en Jaén, se referencian a los proveedores de Toledo y Granada con igual probabilidad

Conjunto P de predicados simples

$P = \{\text{ciudad} = \text{'Toledo'}, \text{ciudad} = \text{'Granada'}\}$

Términos de predicados

$y_1 : \text{ciudad} = \text{'Toledo'} \wedge \text{ciudad} = \text{'Granada'}$

Falso

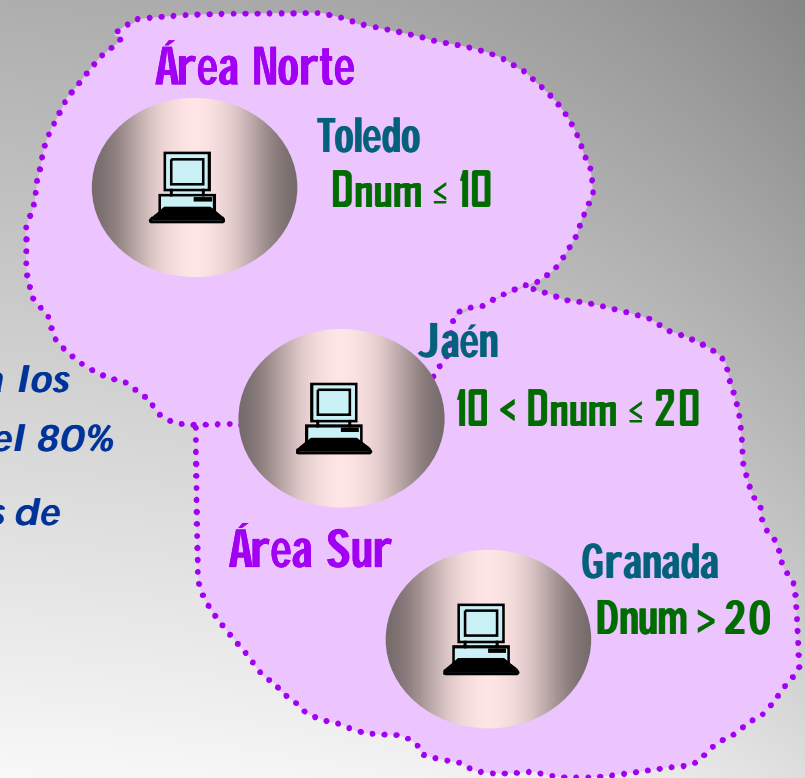


$y_2 : \text{ciudad} = \text{'Toledo'} \wedge \neg \text{ciudad} = \text{'Granada'}$

$y_3 : \neg \text{ciudad} = \text{'Toledo'} \wedge \text{ciudad} = \text{'Granada'}$

Falso

$y_4 : \neg \text{ciudad} = \text{'Toledo'} \wedge \neg \text{ciudad} = \text{'Granada'}$



Fragmentos

$\text{Proveedor}_1 = \text{SL}_{\text{ciudad}=\text{'Toledo'}}(\text{Proveedor})$

$\text{Proveedor}_2 = \text{SL}_{\text{ciudad}=\text{'Granada'}}(\text{Proveedor})$

¿Dónde se asignarían?

Diseño De la fragmentación de una base de datos

Ejemplo general (continuación)

Relación global a fragmentar

Departamento = (Dnum, nombre, área, mgrnum)

+ *Aplicaciones administrativas que se generan en Toledo, para departamentos del área Norte, y en Granada para los del área Sur*

+ *Aplicaciones sobre el trabajo en los departamentos que se generan en cualquier localidad, pero que referencian a departamentos cercanos con mayor probabilidad*

Conjunto P de predicados simples

$P = \{Dnum \leq 10, 10 < Dnum \leq 20, Dnum > 20, \text{área} = \text{'Norte'}, \text{área} = \text{'Sur'}\}$

Términos de predicados

$y_1 : Dnum \leq 10$

$y_2 : 10 < Dnum \leq 20 \wedge \text{área} = \text{'Norte'}$

$y_3 : 10 < Dnum \leq 20 \wedge \text{área} = \text{'Sur'}$

$y_4 : Dnum > 20$

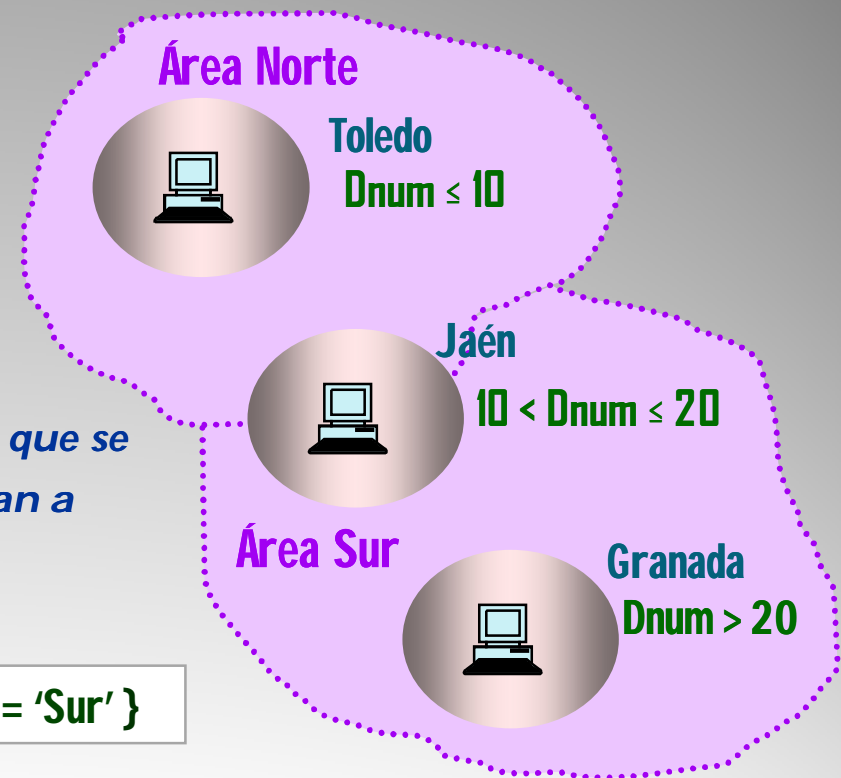
Fragmentos

$\text{Departamento}_1 = SL_{Dnum \leq 10} (\text{Departamento})$

$\text{Departamento}_2 = SL_{10 < Dnum \leq 20 \wedge \text{área} = \text{'Norte'}} (\text{Departamento})$

$\text{Departamento}_3 = SL_{10 < Dnum \leq 20 \wedge \text{área} = \text{'Sur'}} (\text{Departamento})$

$\text{Departamento}_4 = SL_{Dnum > 20} (\text{Departamento})$



¿Dónde se asignarían?

Diseño De la fragmentación de una base de datos

Fragmentación horizontal derivada

Se utiliza para facilitar el producto natural distribuido

Un **producto natural distribuido** es un producto natural entre relaciones fragmentadas horizontalmente

Un producto natural entre dos relaciones globales R y S requiere comparar todas las tuplas de R con todas las tuplas de S



Si distribuido

Comparar todos los fragmentos R_i de R con todos los fragmentos S_j de S



A veces

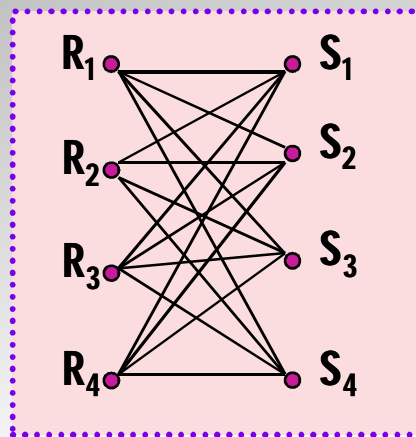
Se puede deducir que algunos productos parciales $R_i \Join S_j$ son intrínsecamente vacíos

¿Cuándo?

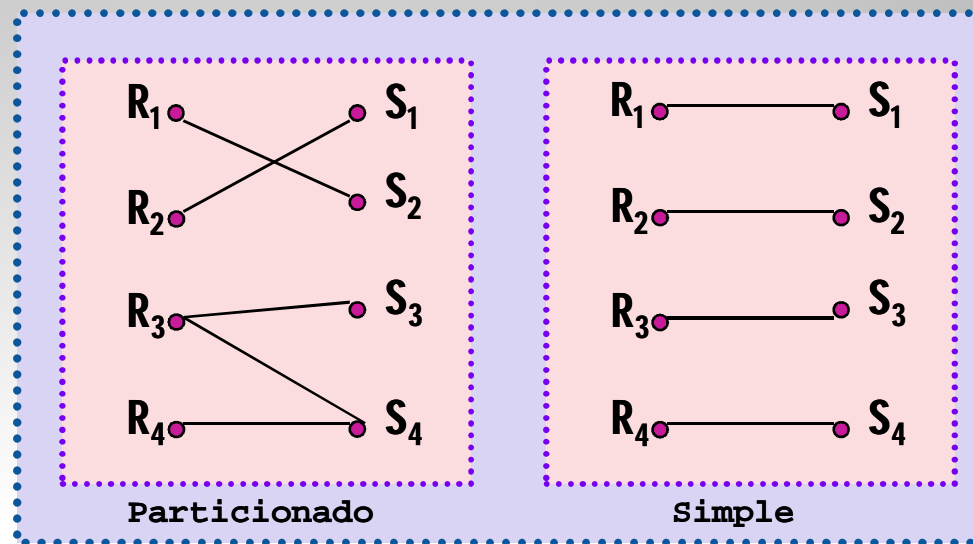
Diseño De la fragmentación de una base de datos

Grafos de producto natural

El grafo de un producto natural distribuido se define como un grafo $G = (N, E)$, donde los nodos N representan fragmentos y las flechas no dirigidas E representan productos naturales no vacíos entre fragmentos



Total



Reducido

Teorema

Sea R una relación global cuyos fragmentos R_i se derivan de la fragmentación de otra relación global S , es decir $R_i = R \bowtie_{JN_F} S_i$. Si se verifican las condiciones de fragmentación completa y disjunta, entonces $R \bowtie_{JN_F} S$ tiene un grafo de producto natural simple

Diseño De la fragmentación de una base de datos

Ejemplo general (continuación)

Relación global a fragmentar

Suministro = (Pnum, Anum, Dnum, cantidad)

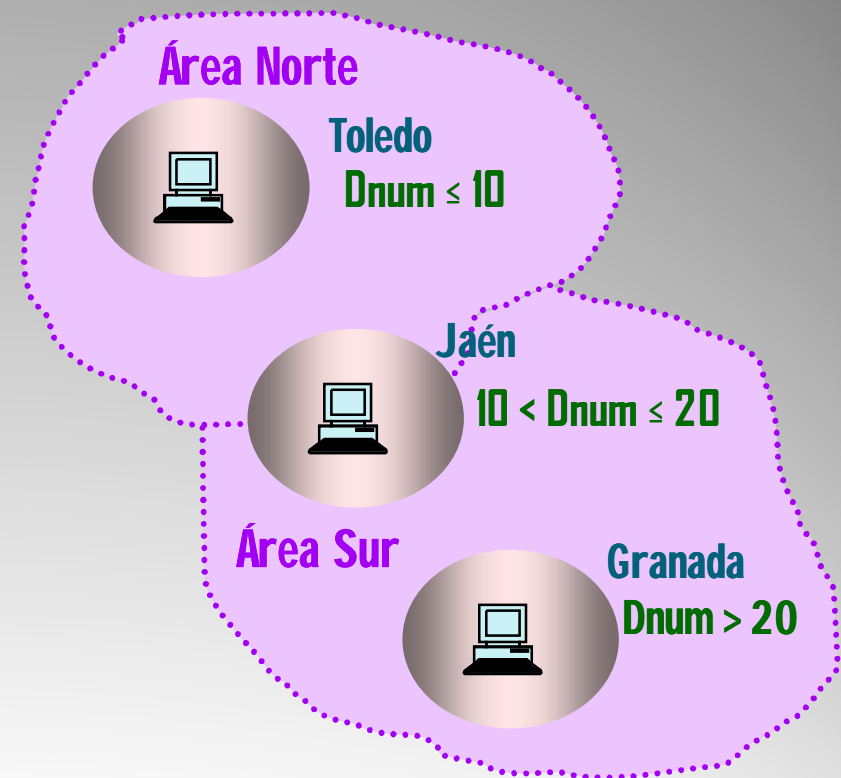
- ✚ Aplicaciones que requieren información sobre los suministros de los proveedores
- ✚ Aplicaciones que requieren información sobre los suministros de los departamentos

Fragmentos

$\text{Suministro}_1 = \text{Suministro}_{\text{SJN}}^{\text{Pnum} = \text{Pnum}_1} \text{Proveedor}_1$
 $\text{Suministro}_2 = \text{Suministro}_{\text{SJN}}^{\text{Pnum} = \text{Pnum}_2} \text{Proveedor}_2$

Fragmentos

$\text{Suministro}_1 = \text{Suministro}_{\text{SJN}}^{\text{Dnum} = \text{Dnum}_1} \text{Departamento}_1$
 $\text{Suministro}_2 = \text{Suministro}_{\text{SJN}}^{\text{Dnum} = \text{Dnum}_2} \text{Departamento}_2$
 $\text{Suministro}_3 = \text{Suministro}_{\text{SJN}}^{\text{Dnum} = \text{Dnum}_3} \text{Departamento}_3$
 $\text{Suministro}_4 = \text{Suministro}_{\text{SJN}}^{\text{Dnum} = \text{Dnum}_4} \text{Departamento}_4$



¿Cuál?

Diseño De La fragmentación de una base de datos

Fragmentación vertical

Agrupar atributos que son referenciados de la **misma manera** por las aplicaciones

Correctitud

Cada atributo de R tiene que estar contenido en, al menos, un fragmento y cada fragmento debe incluir una llave de R o un "identificador de tupla"

Propósito

Identificar los fragmentos R_i tales que muchas aplicaciones puedan ejecutarse utilizando exactamente uno de ellos

Diseño De la fragmentación de una base de datos

Heurísticas de fragmentación

- ✦ Escindir
Las relaciones globales son progresivamente divididas en fragmentos
- ✦ Agrupar
Los atributos son progresivamente añadidos para construir los fragmentos

Tipos de fragmentación vertical

- ✦ Partición
Conjuntos de atributos disjuntos
- ✦ Agrupación
Conjuntos de atributos que pueden solaparse



Introduce replicación
Ventajosa para aplicaciones de sólo lectura y **no conveniente** para aplicaciones de actualización

Diseño De la fragmentación de una base de datos

Ejemplo general (continuación)

Relación global a fragmentar

Empleado = (Enum, nombre, salario, IRPF, mgrnum, Dnum)

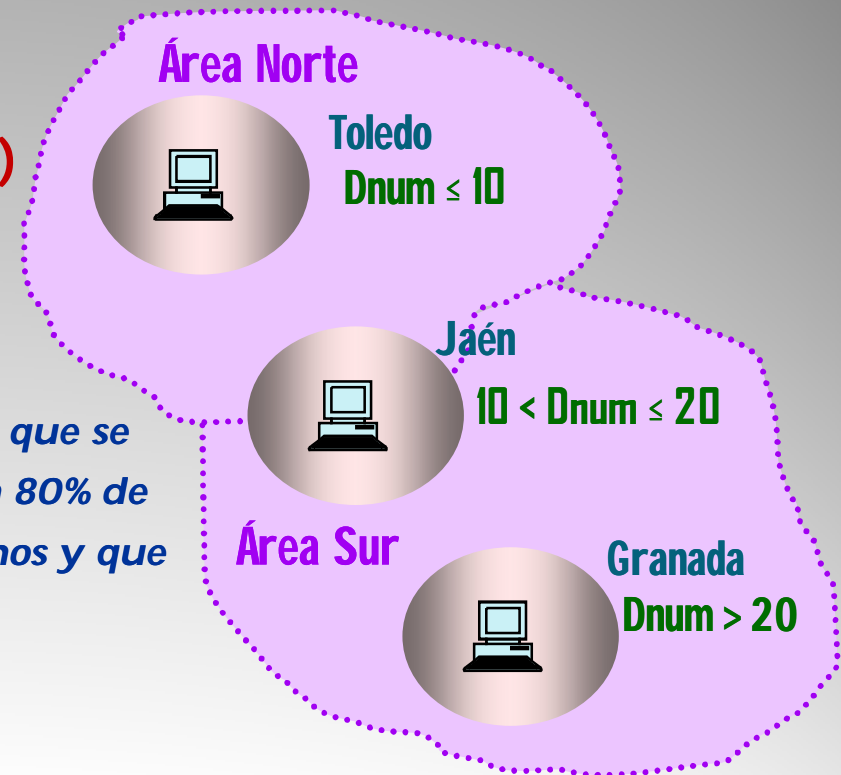
- ✚ *Aplicaciones administrativas que se generan en Granada y que requieren el nombre, salario e IRPF de los empleados*
- ✚ *Aplicaciones sobre el trabajo en los departamentos que se generan en cualquier localidad, referencian con un 80% de probabilidad a empleados de departamentos cercanos y que requieren el nombre, mgrnum y Dnum*

Fragmentos

$\text{Empleado}_1 = \text{PJ}_{\text{Enum, nombre, salario, IRPF}}(\text{Empleado})$

$\text{Empleado}_2 = \text{PJ}_{\text{Enum, nombre, mgrnum, Dnum}}(\text{Empleado})$

¿Sería mejor una partición?



Criterios generales para la asignación de fragmentos

■ Asignación no redundante

Aproximación “más adecuada”

A cada posible asignación se le asocia una medida, seleccionándose la localidad con mejor valor

Problema

Descuida el efecto de asignar un fragmento en una localidad en la que existe otro fragmento relacionado

■ Asignación redundante

Aproximación “todas las localidades beneficiosas”

Determinar el conjunto de localidades en las que el beneficio de asignar una copia es mayor que el coste

Aproximación “replicación adicional”

Determinar la solución al problema no redundante, e introducir progresivamente copias hasta que ninguna réplica sea beneficiosa

Problemas

- ➔ El grado de replicación de cada fragmento es una variable del problema
- ➔ Modelar las aplicaciones de sólo lectura es más complicado

Medida del coste y beneficio de la asignación de fragmentos

Notación

- i es el índice del fragmento
- j es el índice de la localidad
- k es el índice de la aplicación
- f_{kj} frecuencia de activación de la aplicación k en la localidad j
- r_{ki} número de referencias que la aplicación k necesita realizar para recuperar datos del fragmento i
- u_{ki} número de referencias que la aplicación k necesita realizar para actualizar datos del fragmento i
- $n_{kj} = r_{ki} + u_{ki}$

Medida del coste y beneficio de la asignación de fragmentos

Fragmentación horizontal

- Aproximación “más adecuada”

$$B_{ij} = \sum_k f_{kj} n_{ki}$$

El fragmento i se asigna a la localidad j en la que B_{ij} sea máximo

- Aproximación “todas las localidades beneficiosas”

$$B_{ij} = \sum_k f_{kj} n_{ki} - C * \sum_k \sum_{j' \neq j} f_{kj'} u_{ki}$$

C es una constante que mide la proporción entre el coste de actualizar y recuperar ($C \geq 1$)

El fragmento i se asigna a todas las localidades j en las que B_{ij} sea positivo

- Aproximación “replicación adicional”

$$B_{ij} = \sum_k f_{kj} n_{ki} - C * \sum_k \sum_{j' \neq j} f_{kj'} u_{ki} + \beta(d_i)$$

$$\beta(d_i) = (1 - 2^{1-d_i}) F_i$$

d_i = grado de replicación del fragmento i

F_i = beneficio de tener el fragmento i completamente replicado



Beneficio de asignar una copia del fragmento i en términos del aumento de la fiabilidad y disponibilidad

Medida del coste y beneficio de la asignación de fragmentos

Fragmentación vertical

- Aplicaciones A_s y A_t generadas en s o t , que usan, solamente, atributos de R_s o R_t

Evitan una referencia remota

- Aplicaciones A_1 , antes locales a r , que usan, solamente, atributos de R_s o R_t

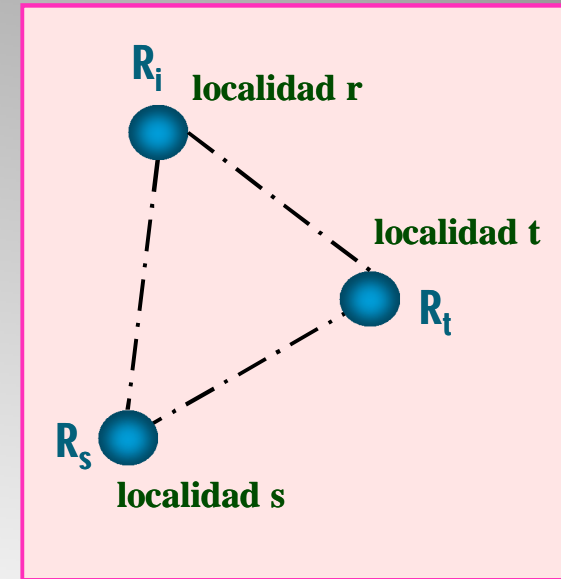
Añaden una referencia remota

- Aplicaciones A_2 , antes locales a r , que usan atributos de R_s y de R_t

Añaden dos referencias remotas

- Aplicaciones A_3 , generadas en localidades diferentes a r , s y t , que usan atributos de R_s o R_t

Añaden una referencia remota



$$B_{ist} = \sum_{k \in A_s} f_{ks} n_{ki} \sum_{k \in A_t} f_{kt} n_{ki} - \sum_{k \in A_1} f_{kr} n_{ki} - 2 \sum_{k \in A_2} f_{ks} n_{kr} - \sum_{k \in A_3} \sum_{j \neq r, s, t} f_{kj} n_{ki}$$

Se realiza la fragmentación vertical de R_i si B_{ist} es positivo