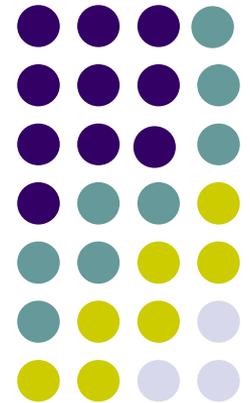


Projeto de Bases de Dados Distribuídas

Pablo Vieira Florentino



Agenda



- Estratégias
- Questões
- Técnicas de Fragmentação Horizontal
 - Primária (3)
 - Derivada
- Corretude da Fragmentação H
- Fragmentação Vertical
- Corretude da Fragmentação V

Problemática da distribuição

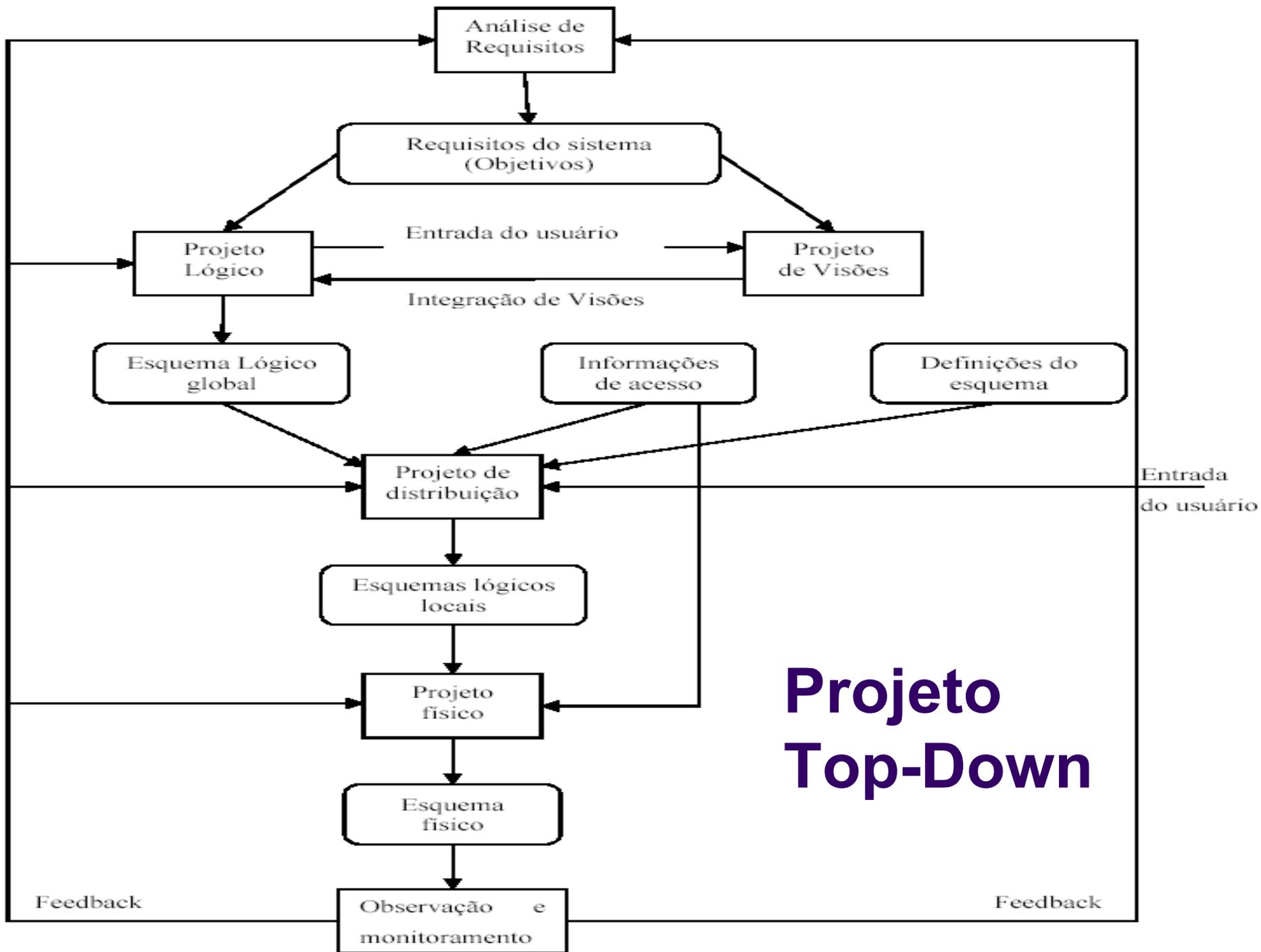


- No âmbito geral:
 - Tomar decisões sobre fragmentação/alocação de dados e programas em sítios conectados por uma rede de computadores assim como projetar a própria rede;
- Em SGBDs Distribuídos,
 - alocação das aplicações e do próprio SGBD distribuído;
 - Fragmentação/alocação das coleções de dados pertencentes à base de dados;

Estratégias



- Top-down
 - Mais indicados quando:
 - o projeto acontece desde o início do desenvolvimento do sistema
 - Sistemas homogêneos
- Bottom-up
 - Quando a base de dados já existe e encontra-se espalhada em sítios com sistemas heterogêneos



Questões

- ✓ Por que fragmentar ?
- Como fragmentar ?
- Quanto fragmentar?
- Como testar a corretude?
- Como alocar ?
- Requisitos sobre a aplicação?



O que considerar ?



- **Informações de acesso**
 - **Frequência dos acessos**
 - **Tipos de acessos**
- **Informações da base de dados**
 - **Relacionamentos**
 - **Cardinalidade**
 - **Tipo dos campos**
 - **Tamanho dos campos**

Fragmentação Horizontal (FH)



- Divide uma relação r separando as suas tuplas em dois ou mais fragmentos
- Assim, a relação r é “particionada horizontalmente” em um número n de subconjuntos: r_1, r_2, \dots, r_n
- Cada tupla da relação r deve pertencer a um dos fragmentos
- A relação r é a união de todos os fragmentos, isto é:
 - $r = r_1 \cup r_2 \cup \dots \cup r_n$

Fragmentação Horizontal Primária (FHP)



- **Relacionamentos**

CONTA →

AGÊNCIA	CONTA	SALDO
Brasília	A-110	1000
Brasília	A-120	500
São Paulo	A-500	500
São Paulo	A-200	300
Brasília	A-150	250
Brasília	A-180	2000

Fragmentação Horizontal Primária



- **Cardinalidades**
- **Natureza dos acessos das aplicações**
 - **Predicados (condições)** : Dada uma relação $R[A_1, A_2, \dots, A_n]$, um predicado simples p_j é
$$p_j : A_i \theta = Value$$
onde $\theta \in \{=, <, \leq, >, \geq, \neq\}$, $Value \in D_i$ e D_i é o domínio de A_i . Assim, para a relação R , podemos definir $Pr = \{p_1, p_2, \dots, p_m\}$
Exemplo :
 - Agencia = "São Paulo"
 - Saldo \geq 500

Fragmentação Horizontal Primária



- **Geração dos Mintermos:** Dados R e

$P_r = \{p_1, p_2, \dots, p_m\}$ define-se $M = \{m_1, m_2, \dots, m_r\}$

como $M = \{ m_i \mid m_i = \bigwedge_{p_j \in P_r} p_j^* \}, 1 \leq j \leq m, 1 \leq i \leq r$

onde $p_j^* = p_j$ ou $p_j^* = \neg(p_j)$.

- Exemplo:

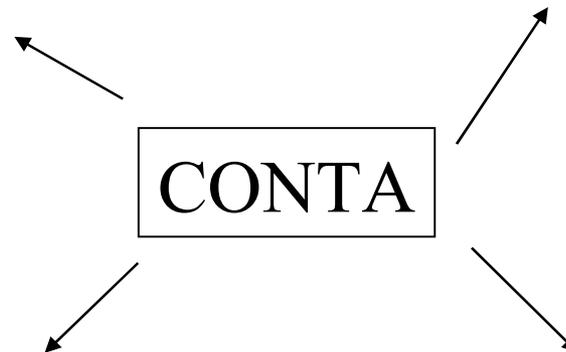
- m_1 : Agência="São Paulo" \wedge Saldo \geq 500
- m_2 : **NOT**(Agência="São Paulo") \wedge Saldo \geq 500
- m_3 : Agência="São Paulo" \wedge **NOT**(Saldo \geq 500)
- m_4 : **NOT**(Agência="São Paulo") \wedge **NOT**(Saldo \geq 500)



Fragmentação Horizontal Primária

AGÊNCIA	CONTA	SALDO
Brasília	A-110	1000
Brasília	A-120	500
Brasília	A-180	2000

AGÊNCIA	CONTA	SALDO
Brasília	A-150	250



AGÊNCIA	CONTA	SALDO
São Paulo	A-500	500

AGÊNCIA	CONTA	SALDO
São Paulo	A-200	300

Fragmentação Horizontal Derivada (FHD)



- **Fragmentação guiada pela FHP de uma outra coleção de dados relacionada**
 - Usada para facilitar as operações de junção e navegação entre fragmentos
 - Baseada em relacionamentos “Owner-Member”
 - Tomemos duas coleções A e B, com A relacionada a B.
 - Seja B o “owner” do relacionamento e A o “member”
 - Se B sofreu uma FHP, então A pode sofrer FHD de acordo com a FHP de B



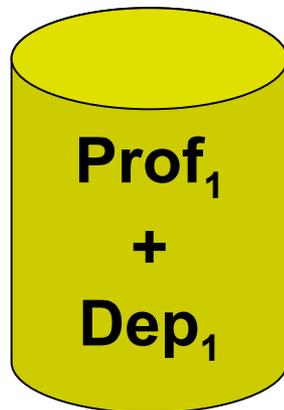
Fragmentação Horizontal Derivada (FHD)



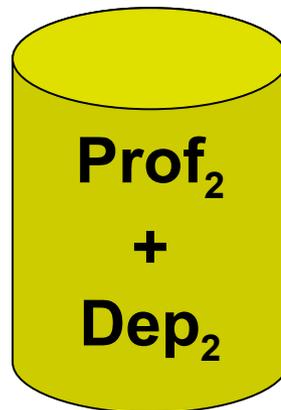
- **Exemplo:**

- Prof1 = Professor ~~X~~ Dep1
- Prof2 = Professor ~~X~~ Dep2
- Prof3 = Professor ~~X~~ Dep3

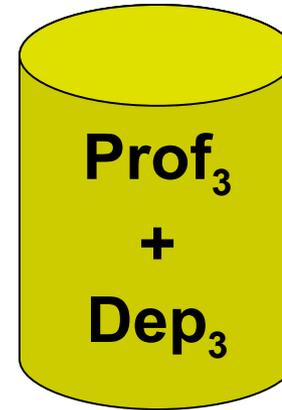
Sítio 1



Sítio 2



Sítio 3



Fragmentação Vertical (FV)



- Mais antiga
- Tem sido estudada no contexto de BD centralizados
 - Metodologia para projeto de SBDD
 - Agrupamento físico (Clustering)
- Mais complexa que a FH por oferecer mais alternativas
 - Pode haver sobreposição de fragmentos
 - Chave da relação presente em todos os fragmentos
 - Avaliação das afinidades dos atributos

Fragmentação Vertical (FV)



- Divide uma relação r organizando os seus atributos em dois ou mais fragmentos
- Assim, a relação r é “particionada verticalmente” em um número n de subconjuntos: r_1, r_2, \dots, r_n
- Cada atributo da relação r deve pertencer a um dos fragmentos
- A relação r é a junção de todos os fragmentos, isto é:
 - $r = r_1 \times r_2 \times \dots \times r_n$

Fragmentação Vertical (FV)



- Coletando informações sobre acessos das aplicações
 - Afinidade dos atributos
 - Medida que indica quão “próximos” estão os atributos em termos de uso/acesso
 - Peso dos acessos aos atributos
 - Dado um conjunto de consultas $Q = \{q_1, q_2, \dots, q_q\}$ que fará acesso à relação $R[A_1, A_2, \dots, A_n]$,

$$use(q_i, A_j) = \begin{cases} 1 & \text{se o atributo } A_j \text{ é usado pela consulta } q_i \\ 0 & \text{otherwise} \end{cases}$$

Fragmentação Vertical (FV)



- Consideremos a seguinte coleção Conta'

AGÊNCIA	CONTA	SALDO	GERENTE
---------	-------	-------	---------

e um conjunto Q de consultas que faz acesso à mesma:

- q_1 : **SELECT** Saldo **FROM** Conta **WHERE** Agencia=Value
- q_2 : **SELECT** Conta,Saldo **FROM** Conta
- q_3 : **SELECT** Conta **FROM** Conta **WHERE** Gerente=Value
- q_4 : **SELECT SUM**(Saldo) **FROM** Conta **WHERE** Gerente=Value

Fragmentação Vertical (FV)



- Sendo A_1 = Agencia, A_2 = Conta, A_3 = Saldo, A_4 = Gerente
 - q_1 : **SELECT** Saldo **FROM** Conta **WHERE** Agencia=Value
 - q_2 : **SELECT** Conta,Saldo **FROM** Conta
 - q_3 : **SELECT** Conta **FROM** Conta **WHERE** Gerente=Value
 - q_4 : **SELECT SUM**(Saldo) **FROM** Conta **WHERE** Gerente=Valuepode-se definir a matriz de acesso

	A_1	A_2	A_3	A_4
q_1	1	0	1	0
q_2	0	1	1	0
q_3	0	1	0	1
q_4	0	0	1	1

Fragmentação Vertical (FV)



- Após isso, deve-se definir a matriz de afinidade entre os atributos (Matriz AA)
 - Contabilizar a frequência de cada uma das consultas em todos os sítios. Suponhamos então as seguintes frequências:

$$F(q1)=45 \quad F(q2)=5 \quad F(q3)=75 \quad F(q4)=3$$

- Calcular a afinidade dos atributos dois a dois
 - $aff(A_i, A_j)$ = frequência de todas as consultas que fazem acesso a A_i e A_j

• *Exemplo:*

$$aff(A_1, A_3) = F(q1) = 45$$

Matriz AA →

	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

Fragmentação Vertical (FV)



- Definida a Matriz AA de afinidade, é necessário tentar reorganizar a ordem dos atributos no sentido de agrupá-los, indicando subconjuntos de atributos que formarão os fragmentos da relação
- Algoritmo BEA (Bond Energy Algorithm): procura maximizar a medida de afinidade global dos atributos
 - Realiza troca nas posições das colunas que formam a Matriz AA
 - Uma das colunas deve ser posicionada inicialmente
 - Cada coluna deve ser posicionada no local onde aconteça maior contribuição para a medida de afinidade

$$\underline{cont(A_i, A_k, A_j) = 2bond(A_i, A_k) + 2bond(A_k, A_j) - 2bond(A_i, A_j)}$$

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

- Ao fim, as linhas devem ser ordenadas como as colunas

Fragmentação Vertical (FV)



- Exemplo:

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

- Ordenando (0-3-1) :
 $cont(A_0, A_3, A_1) = 2bond(A_0, A_3) + 2bond(A_3, A_1) - 2bond(A_0, A_1)$
 $= 2 * 0 + 2 * 4410 - 2 * 0 = 8820$
- Ordenando (1-3-2) :
 $cont(A_1, A_3, A_2) = 2bond(A_1, A_3) + 2bond(A_3, A_2) - 2bond(A_1, A_2)$
 $= 2 * 4410 + 2 * 890 - 2 * 225 = 10150$
- Ordenando (2-3-4) :
 $cont(A_2, A_3, A_4) = 1780$

Fragmentação Vertical (FV)



- Desta forma, teríamos:

$$\begin{array}{c} A_1 \quad A_3 \quad A_2 \\ \left[\begin{array}{ccc} 45 & 45 & 0 \\ 0 & 5 & 80 \\ 45 & 53 & 5 \\ 0 & 3 & 75 \end{array} \right] \end{array}$$

Após realizar todas as
permutas

$$\begin{array}{c} A_1 \quad A_3 \quad A_2 \quad A_4 \\ \left[\begin{array}{cccc} 45 & 45 & 0 & 0 \\ 45 & 53 & 5 & 3 \\ 0 & 5 & 80 & 75 \\ 0 & 3 & 75 & 78 \end{array} \right] \end{array}$$

Fragmentação Vertical (FV)

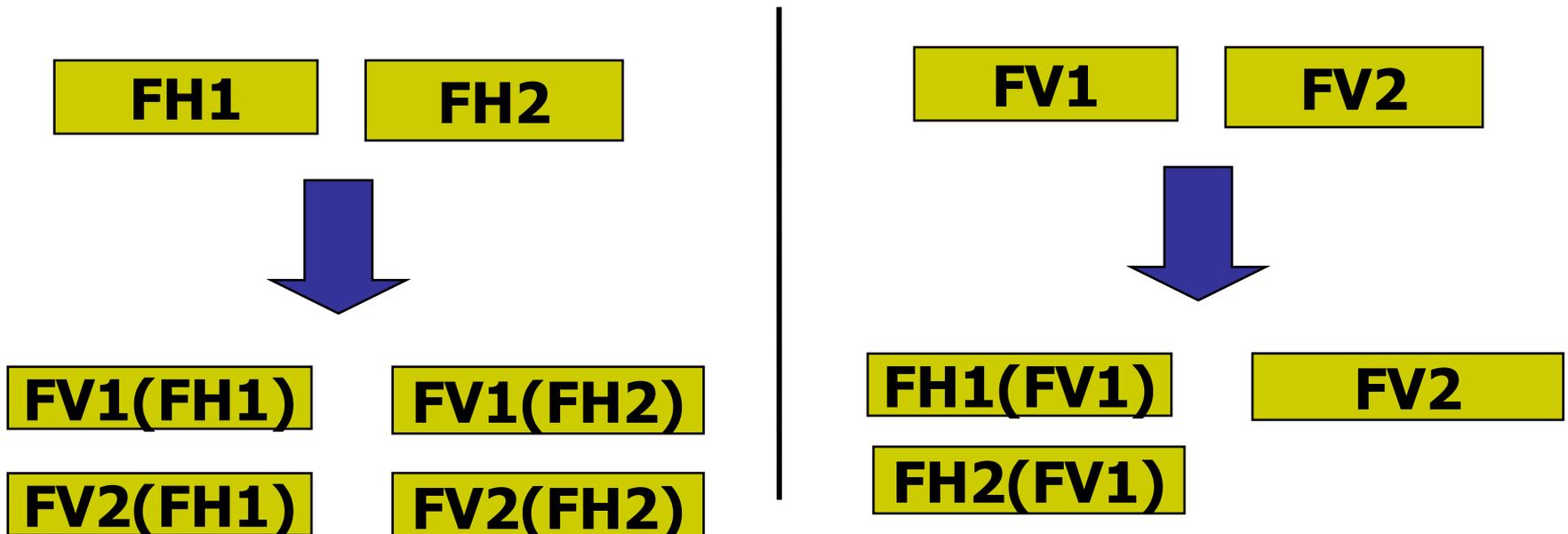


- Problemas:
 - Grande complexidade
 - $O(m^2)$
 - Agrupamentos centrais
 - Etc ...
- Alternativa:
 - Bom senso x Características da Aplicação
 - Exemplos

Fragmentação Híbrida



- Aplicação de ambas as técnicas sobre a mesma classe
 - Heurística: Aplicar 1o a FV, depois a FH.
 - Objetivo: gerar menos fragmentos



Corretude da Fragmentação



- Completeness
 - A decomposição de uma relação R em fragmentos R_1, R_2, \dots, R_n é considerada completa se, e somente se, cada item de dados em R pode também ser encontrado em algum R_i
- Reconstruction
 - Se uma relação R é decomposta nos fragmentos R_1, R_2, \dots, R_n , então deve existir algum operador relacional W tal que
$$R = W_{1 \leq i \leq n} R_i$$
- Disjointness
 - Se uma relação R é decomposta nos fragmentos R_1, R_2, \dots, R_n , e o item de dados di está em R_j , então di não deve estar presente em qualquer outro fragmento R_k ($k \neq j$).

Alocação



- **Alternativas**

- Não-replicado

- Cada fragmento de R reside em somente um sítio

- Replicado

- Totalmente replicado : cada fragmento em cada sítio
- Parcialmente replicado : cada fragmento em alguns dos sítios

- Regra Básica (Rule of thumb):

Se $\frac{\text{Num}(\textit{read - only queries})}{\text{Num}(\textit{update queries})} \geq 1$

replicação é vantajosa, caso contrário,

pode gerar problemas e diminuir o throughput do sistema

Alocação – Considerações sobre replicação



- Aumento de paralelismo: outros sites podem processar as consultas em paralelo. Com mais replicação, aumentam as chances do sítio que está realizando a transação possuir o fragmento F_i , contribuindo assim com a minimização do movimento dos dados entre os sítios

Alocação – Considerações sobre replicação



- Aumento do overhead para atualização: o sistema deverá assegurar que todas as réplicas do fragmento F_i sejam consistentes, propagando a atualização a todas as réplicas. Sempre que houver atualização, ela deve ser propagada a todos os sítios que contenham réplicas, evitando a inconsistência. Por exemplo: o saldo de uma conta bancária com um valor, possuindo um outro valor em uma consulta feita por outra agência, sendo ambas de um mesmo banco.



Alocação

- Alocação de Arquivos (FAP)

VS

Alocação de Bases de Dados (DAP):

- Fragmentos não são arquivos individuais
 - relacionamentos têm que ser mantidos
- Acesso às bases de dados é mais complicada
 - Grande dependência entre alocação e processamento de consultas
- Custo de manutenção da integridade dos dados deve ser considerado
- Custo do controle de concorrência deve ser considerado



Requisitos para Alocação

- Definição do problema

Dados

$F = \{F1, F2, \dots, Fn\}$ fragmentos

$S = \{S1, S2, \dots, Sm\}$ sítios da rede

$Q = \{q1, q2, \dots, qq\}$ aplicações

Encontre a distribuição "ótima" de F para S .

- Principais pontos para otimização
 - Custo mínimo
 - Comunicação + armazenamento + processamento (leitura & atualização)
 - Custo em termos de tempo
 - Desempenho
 - Tempo de Resposta e/ou throughput
 - Restrições
 - Por sítio (armazenamento & processamento)



Requisitos para Alocação

- Base de Dados
 - seletividade dos fragmentos
 - Tamanho de cada fragmento
- Informações sobre a Aplicação
 - Tipos e quantidade de acessos
 - Número de acessos de leitura de uma consulta a um fragmento
 - Número de acessos de atualização de um comando a um fragmento
 - Localização dos acessos
 - Uma matriz indicando quais consultas atualizam quais fragmentos
 - Uma outra matriz, similar, contendo somente acessos de leitura
 - Sítio de origem de cada acesso



Requisitos para Alocação

- Sistema Local (por sítio)
 - Custo unitário de armazenamento de dados em um sítio
 - Custo unitário de processamento em um sítio
- Rede de Comunicação
 - Largura de Banda
 - Custo de comunicação por frame entre dois sítios
 - Tamanho do frame de dados
 - Latência
 - Sobrecarga de comunicação

Princípios para cálculo do Modelo de Alocação



- Forma Geral
 - Objetivo: Minimizar o **Custo Total**, que seria formado por

$\sum_{\text{todas consultas}}$ custo de processamento da consulta

+

$\sum_{\text{todos sítios}} \sum_{\text{todos fragmentos}}$ custo de armazenamento de um
fragmento em um sítio

Princípios para cálculo do Modelo de Alocação



- Custo de Armazenamento (de um fragmento F_i num sítio S_k)

(Custo unitário de armazenamento em S_k) * (tamanho de F_i)

- Custo de Processamento de consulta (p/ uma determinada consulta)

(Custo de processamento) + (Custo de transmissão)

Modelo de Alocação



- Problema de grande complexidade
 - Alguns Métodos
 - FAP: é NP-Completo
 - DAP: é NP-Completo
 - Podem ser utilizadas algumas heurísticas, como:
 - Problema da Mochila
 - Técnicas de branch-and-bound
 - Fluxo de redes