CSE 4125: Distributed Database Systems
Chapter – 4
(Part – D)

Distributed Database Design

The Design of Vertical Fragmentation

Vertical Fragmentation

- ✓ Partitioning the attributes of a relation into a set of smaller relations.

  So that many of the applications will run on only one fragment.

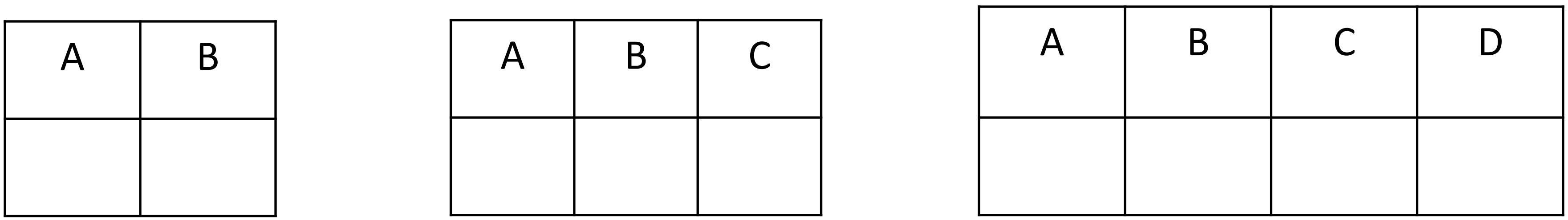
- ✓ Vertical Fragmentation can be done in the following ways:

  -- Clustering*:* sets can be overlapped

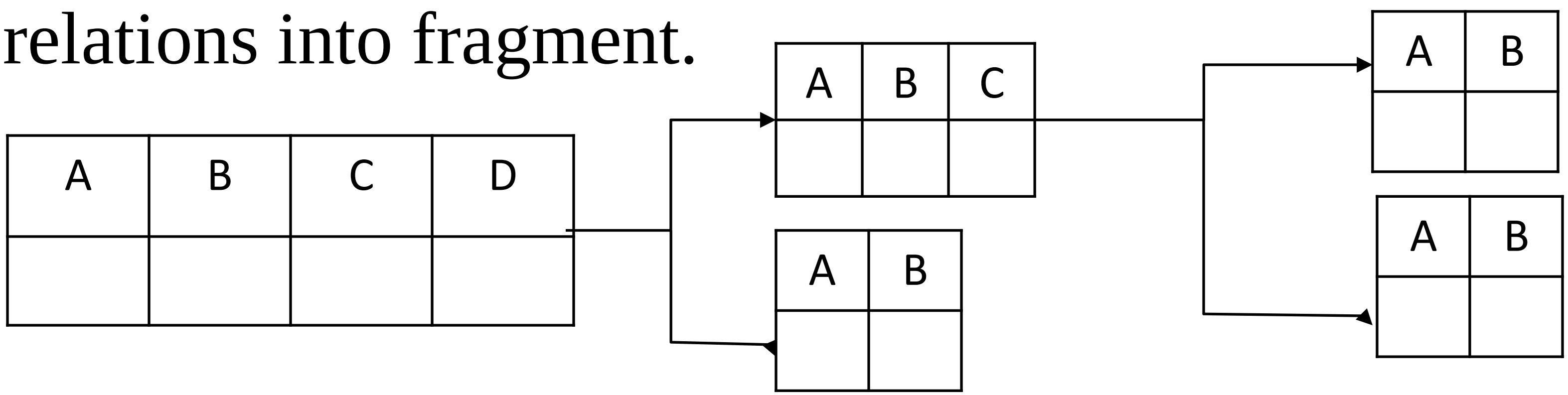
  -- Partitioning*:* sets must be disjoint.

Approaches:

Grouping (clustering): Progressively assigning each attribute to constitute clusters.



Splitting (partitioning): Progressively splitting global relations into fragment.



Bond Energy Algorithm (BEA)

Steps:

1. Attribute Usage Matrix
2. Attribute Affinity Matrix
3. Clustered Affinity Matrix
4. Partitioning

Bond Energy Algorithm (BEA) Example

PROJ

PNO	PNAME	BUDGET	LOC
P1	Instrumental	150,000	Montreal
P2	Database Dev	135,000	New York
P3	CAM/CAD	250,000	New York
P4	Maintenance	310,000	Orlando

Consider the following 4 queries for relation PROJ, where PNO is the primary key column of the table.

q_1: **SELECT** BUDGET
FROM PROJ
WHERE PNO=Value

q_2: **SELECT** PNAME,BUDGET
FROM PROJ

q_3: **SELECT** PNAME
FROM PROJ
WHERE LOC=Value

q_4: **SELECT** **SUM**(BUDGET)
FROM PROJ
WHERE LOC=Value

Assume that, **PROJ** relation is located in three different sites. The access frequency of each query for each site is stated below –

	S1	**S2**	**S3**
q1	15	20	10
q2	5	0	0
q3	25	25	25
q4	3	0	0

FM

Using the Bond Energy Algorithm, group the columns of the table and after that split the columns vertically at required position with the help of goal function.

Note: You should show Attribute Affinity Matrix, Clustered Affinity Matrix, and the calculation for each of the ordering. Take the first two columns as the starting bonding state.

Step 1: Attribute Usage Matrix

Let, A1 = PNO, A2 = PNAME, A3 = BUDGET, A4 = LOC

	A1	A2	A3	A4
Q1	1	0	1	0
Q2	0	1	1	0
Q3	0	1	0	1
Q4	0	0	1	1

2

1

q_1: **SELECT** BUDGET
FROM PROJ
WHERE PNO=Value

q_2: **SELECT** PNAME,BUDGET
FROM PROJ

q_3: **SELECT** PNAME
FROM PROJ
WHERE LOC=Value

q_4: **SELECT** **SUM**(BUDGET)
FROM PROJ
WHERE LOC=Value

Step 2: Attribute Affinity Matrix

3

	A1	A2	A3	A4
A1	45	0	45	0
A2	0	80	5	75
A3	45	5	53	3
A4	0	75	3	78

1

	S1	S2	S3	SUM
Q1	15	20	10	45
Q2	5	0	0	5
Q3	25	25	25	75
Q4	3	0	0	3

2

	A1	A2	A3	A4
Q1	1	0	1	0
Q2	0	1	1	0
Q3	0	1	0	1
Q4	0	0	1	1

Step 3: Clustered Affinity Matrix

Consider the following AA matrix and the corresponding CA matrix where A_1 and A_2 have been placed. **Place $\boldsymbol{A_3}$:**

AA =

	A1	A2	A3	A4
A1	45	0	45	0
A2	0	80	5	75
A3	45	5	53	3
A4	0	75	3	78

Starting Bonding State

CA =

	A1	A2		
A1	45	0		
A2	0	80		
A3	45	5		
A4	0	75		

Global Affinity Measure

New attribute always will be in middle

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

Contribution Function

$$bond(A_x, A_y) = \text{SUMMATION(For all rows (Ax * Ay))}$$

Example –

$$cont(A_1, A_4, A_2) = 2bond(A_1, A_4) + 2bond(A_4, A_2) - 2bond(A_1, A_2)$$

$$bond(A_1, A_4) = 45*0 + 0*75 + 45*3 + 0*78 = 135$$

$$bond(A_4, A_2) = 11865$$

$$bond(A_1, A_2) = 225$$

	A1	A2	A3	A4
A1	45	0	45	0
A2	0	80	5	75
A3	45	5	53	3
A4	0	75	3	78

$$cont(A_1, A_4, A_2) = 2*135 + 2*11865 - 2*225 = 23550$$

Step 3: Clustered Affinity Matrix

Consider the following AA matrix and the corresponding CA matrix where A_1 and A_2 have been placed. **Place A_3:**

Starting Bonding State

AA =

	A1	A2	A3	A4
A1	45	0	45	0
A2	0	80	5	75
A3	45	5	53	3
A4	0	75	3	78

CA =

	A1	A2		
A1	45	0		
A2	0	80		
A3	45	5		
A4	0	75		

Ordering (0-3-1) :

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

Ordering (1-3-2) :

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

Ordering (2-3-4) :

$$cont(A_2,A_3,A_4) = 2bond(A_2, A_3)+2bond(A_3, A_4)-2bond(A_2, A_4)$$
$$= 2*890 + 2*0 - 2*0 = 1780$$

Step 3: Continued

Therefore, the *CA* matrix has to form

	A1	A3	A2	
A1	45	45	0	
A2	0	5	80	
A3	45	53	5	
A4	0	3	75	

Similarly, Now for placing A4 do the calculations. You must have to show the calculation in the exam.

Place A4: All possible orderings will be (0-4-1), (1-4-3), (3-4-2), (2-4-5)

(2-4-5) ordering will be the highest value, means A4 is after A2.

Step 3: Continued

When A_4 is placed, the final form of the CA matrix (**after column organization**) is

	A1	A3	A2	A4
A1	45	45	0	0
A2	0	5	80	75
A3	45	53	5	3
A4	0	3	75	78

The final form of the CA matrix (**after row organization**) is

	A1	A3	A2	A4
A1	45	45	0	0
A3	45	53	5	3
A2	0	5	80	75
A4	0	3	75	78

	A1	A3	A2	A4
A1	45	45	0	0
A3	45	53	5	3
A2	0	5	80	75
A4	0	3	75	78

Clustered Affinity Matrix (CA)

Cluster 1: A_1 & A_3

Cluster 2: A_2 & A_4

What if the clustered affinity matrix looks like this? →

	A1	A3	A2	A4
A1	45	0	75	45
A3	0	53	5	0
A2	75	5	3	0
A4	45	0	0	78

	A1	A3	A2	A4
A1	45	0	75	45
A3	0	53	5	0
A2	75	5	3	0
A4	45	0	0	78

First Left Rotate:

Column

1

	A3	A2	A4	A1
A1	0	75	45	45
A3	53	5	0	0
A2	5	3	0	75
A4	0	0	78	45

Row

2

	A3	A2	A4	A1
A3	53	5	0	0
A2	5	3	0	75
A4	0	0	78	45
A1	0	75	45	45

Clustering Summary (Steps 1,2,3)

- We need AUM that reflects the query-attribute relationship
- AUM and FM are used to make AA
- Global Affinity Measure is used to establish the clusters of attributes
- Stronger affinities attributes and weaker ones are grouped in CA

Step 4: Partitioning

We define -

TQ = set of applications that access only TA

BQ = set of applications that access only BA

OQ = set of applications that access both TA and BA

<table>
<tr><th></th><th>A1</th><th>A3</th><th>A2</th><th>A4</th></tr>
<tr><th>A1</th><td colspan="2" rowspan="2">TA</td><td colspan="2" rowspan="2"></td></tr>
<tr><th>A3</th></tr>
<tr><th>A2</th><td colspan="2" rowspan="2"></td><td colspan="2" rowspan="2">BA</td></tr>
<tr><th>A4</th></tr>
</table>

Step 4: Partitioning

We define -

CTQ = number of accesses to attributes by applications that access only TA

CBQ = number of accesses to attributes by applications that access only BA

COQ = number of accesses to attributes by applications that access both TA and BA

Step 4: Partitioning

Goal Function –

Find the point z along the diagonal that maximizes

$$z = (CTQ * CBQ) – (COQ * COQ)$$

	A1	**A3**	**A2**	**A4**
A1	45	45	0	0
A3	45	53	5	3
A2	0	5	80	75
A4	0	3	75	78

Step 4: Partitioning

Goal Function –

Find the point z along the diagonal that maximizes

$$z = (CTQ * CBQ) - (COQ * COQ)$$

Setting 1

	A1	**A3**	**A2**	**A4**
A1	45	45	0	0
A3	45	53	5	3
A2	0	5	80	75
A4	0	3	75	78

Setting 2

	A1	**A3**	**A2**	**A4**
A1	45	45	0	0
A3	45	53	5	3
A2	0	5	80	75
A4	0	3	75	78

Setting 3

	A1	**A3**	**A2**	**A4**
A1	45	45	0	0
A3	45	53	5	3
A2	0	5	80	75
A4	0	3	75	78

Step 4: Partitioning

Setting 1

	A1	A3	A2	A4
A1	45	45	0	0
A3	45	53	5	3
A2	0	5	80	75
A4	0	3	75	78

AUM

	A1	A2	A3	A4
Q1	1	0	1	0
Q2	0	1	1	0
Q3	0	1	0	1
Q4	0	0	1	1

FM

	S1	S2	S3
q1	15	20	10
q2	5	0	0
q3	25	25	25
q4	3	0	0

TQ = {}

BQ = {q2, q3, q4}

OQ = {q1}

CTQ = 0

CBQ = 5+0+0+25+25+25+3+0+0 = 83

COQ = 15 + 20 + 10 = 45

Z1 = (CTQ * CBQ) – (COQ*COQ) = (0 * 83) – (45 * 45) = -2025

Step 4: Partitioning

Setting 2

	A1	A3	A2	A4
A1	45	45	0	0
A3	45	53	5	3
A2	0	5	80	75
A4	0	3	75	78

AUM

	A1	A2	A3	A4
Q1	1	0	1	0
Q2	0	1	1	0
Q3	0	1	0	1
Q4	0	0	1	1

FM

	S1	S2	S3
q1	15	20	10
q2	5	0	0
q3	25	25	25
q4	3	0	0

TQ = {q1}

BQ = {q3}

OQ = {q2, q4}

CTQ = 15+20+10 = 45

CBQ = 25 + 25+ 25 = 75

COQ = 5+0+0+3+0+0 = 8

Z2 = (CTQ * CBQ) – (COQ*COQ) = (45 * 75) – (8 * 8) = 3311

Step 4: Partitioning

Setting 3

	A1	A3	A2	A4
A1	45	45	0	0
A3	45	53	5	3
A2	0	5	80	75
A4	0	3	75	78

AUM

	A1	A2	A3	A4
Q1	1	0	1	0
Q2	0	1	1	0
Q3	0	1	0	1
Q4	0	0	1	1

FM

	S1	S2	S3
q1	15	20	10
q2	5	0	0
q3	25	25	25
q4	3	0	0

TQ = {q1,q2}

BQ = {}

OQ = {q3, q4}

CTQ = 15+20+10+5+0+0 = 50

CBQ = 0

COQ = 25+25+25+3+0+0 = 78

Z3 = (CTQ * CBQ) – (COQ*COQ) = (50 * 0) – (78 * 78) = -6084

Step 4: Partitioning

Goal Function z2 is maximum with setting 2.

Setting 2

	A1	A3	A2	A4
A1	45	45	0	0
A3	45	53	5	3
A2	0	5	80	75
A4	0	3	75	78

Two Fragments:

PROJ1 = {A1, A3}
= {PNO, BUDGET}

PROJ2 = {A1, A2, A4}
= {PNO, PNAME, LOC}

PNO is the primary key

Vertical Fragmentation

Introduces ***replication.***

– Tuple identifier.

Convenient for read-only application.

– Why?

Not convenient for update application.

– Why?

Answer to why:

Let us consider what happens when two fragments $\boldsymbol{R_1}$ and $\boldsymbol{R_2}$ are overlapping ;i.e., there exists a set of attributes ***I*** which belong to both $\boldsymbol{R_1}$ and $\boldsymbol{R_2}$. Assume that $\boldsymbol{R_1}$ and $\boldsymbol{R_2}$ are at **sites 1 and 2**.

Then **read applications** at site 1, using attributes of ***I*** together with other attributes of $\boldsymbol{R_1}$, are local to site 1; likewise, read applications at site 2, using attributes of ***I*** together with other attributes of $\boldsymbol{R_2}$, are local to site 2.

However, update applications which-change the value of attributes of ***I*** must reference them at both sites.

Exercise

Consider the applications "AP1", "AP2", "AP3" and "AP4" as shown below. These applications work on the **EMP** relation defined as **EMP(EmpID, Name, Loc, Dept),** where **EmpID** is the primary key column of the table.

AP1: SELECT **EmpID** FROM **EMP** WHERE **Dept = "Payroll"**

AP2: SELECT **Dept** FROM **EMP**

AP3: UPDATE **EMP** SET **Loc = "Chittagong"** WHERE **Name = "Kalam"**

AP4: UPDATE **EMP** SET **Loc = "Comilla"** WHERE **EmpID = 109288**

Assume that there is only **one site** and the access frequency of AP1, AP2, AP3 and AP4 is 3, 7, 4, 3 respectively.

Using the Bond Energy Algorithm, group the columns of the table and after that split the columns vertically at required position with the help of goal function.

Note: You should show Attribute Affinity Matrix, Clustered Affinity Matrix, and the calculation for each of the ordering, goal function. Take the first two columns as the starting bonding state.