5.17) Considering that 60% of operations are 'update' on PNO and RESP, while DUR is not updated, it is reasonable to suggest a vertical fragmentation into two fragments, one containing the update attributes and the other those that do not update, such that:

$ASG_A(ENO, PNO, RESP)$

$ASG_B(ENO, PNO, DUR)$

Reason for this being is that update operations are more expensive than read operations.

Ignoring ~~fragmentation~~ storage costs, the optimal fragmentation strategy would be one in which only those attributes accessed by the queries executed on each site are present. For example, at site 1, only q1 is executed therefore we just need those entries for which DUR = 24 and TITLE = 'Programmer'. Fragmenting accordingly results in:

ASG: $ASG_1(ENO, PNO, RESP)$: DUR = 24 $\quad$ $ASG_2(ENO, PNO, RESP)$: DUR <> 24

$ASG_3(ENO, PNO, DUR=24)$ $\quad$ $ASG_4(ENO, PNO, DUR <> 24)$

EMP: $EMP_1(ENO, ENAME, TITLE)$: TITLE = "Programmer"

$EMP_2(ENO, ENAME, TITLE)$: TITLE <> "Programmer"

Putting this together, we get

site 1 (q_1: 10)	EMP_1, ASG_1, ASG_3
site 2 (q_1: 20) (q_2: 20)	EMP_1, EMP_2, ASG_1, ASG_2, ASG_3, ASG_4
site 3 (q_2: 20)	ASG_3, ASG_4

DDMS HW1 2019280366 Albert Sebastia Millan Truyols

5.2)

p_1: RESP = 'Manager'
p_2: RESP = 'Analyst'
p_3: RESP = 'Consultant'
p_4: RESP = 'Engineer'
p_5: RESP = 'Programmer'

p_6: DUR < 20
p_7: DUR ≥ 20

m_1: RESP = 'Manager' ∧ DUR < 20
m_2: RESP = 'Manager' ∧ DUR ≥ 20
m_3: RESP = 'Analyst' ∧ DUR < 20
m_4: RESP = 'Analyst' ∧ DUR ≥ 20
m_5: RESP = 'Consultant' ∧ DUR < 20
m_6: RESP = 'Consultant' ∧ DUR ≥ 20
m_7: RESP = 'Engineer' ∧ DUR < 20
m_8: RESP = 'Engineer' ∧ DUR ≥ 20
m_9: RESP = 'Programmer' ∧ DUR < 20
m_{10}: RESP = 'Programmer' ∧ DUR ≥ 20

ENO	PNO	RESP	DUR
E1	P1	Manager	12

ENO	PNO	RESP	DUR
E5	P2	Manager	24
E6	P4	Manager	48
E8	P3	Manager	40

ENO	PNO	RESP	DUR
E2	P2	Analyst	6

ENO	PNO	RESP	DUR
E2	P1	Analyst	24

ENO	PNO	RESP	DUR
E3	P3	Consultant	10

ENO	PNO	RESP	DUR
E3	P4	Engineer	48
E7	P3	Engineer	36

ENO	PNO	RESP	DUR
E4	P2	Programmer	18

5.8) A_1 = EMP.ENO

A_2 = EMP. ENAME

A_3 = EMP.TITLE

A_4 = ASG.ENO

A_5 = ASG.PNO

A_6 = ASG.RESP

A_7 = ASG. DUR

ASG.ENO and ASG.PNO ~~[illegible]~~ are the composite key.

Use Matrix

	A_1	A_2	A_3	A_4	A_5	A_6	A_7
q_1	1	1	1	1	1	1	1
q_2	0	0	0	1	0	0	1

Access Frequency Matrix

	S_1	S_2	S_3
q_1	10	20	0
q_2	0	20	10

Attribute Affinity Matrix (AA)

A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7
A_1	30	30	30	30	30	30	30
A_2	30	30	30	30	30	30	30
A_3	30	30	30	30	30	30	30
A_4	30	30	30	60	30	30	60
A_5	30	30	30	30	30	30	30
A_6	30	30	30	30	30	30	30
A_7	30	30	30	60	30	30	60

Cluster Affinity Matrix (CA) using BEA

	A_7	A_4	A_1	A_2	A_3	A_5	A_6
A_7	60	60	30	30	30	20	30
A_4	60	60	30	30	20	30	20
A_1	30	30	30	30	30	30	30
A_2	30	30	30	30	20	20	30
A_3	30	30	80	20	30	30	30
A_5	30	20	30	20	30	30	30
A_6	20	30	30	30	30	20	30

∴ ASG_1

ENO | PNO | RESP

∴ ASG_2

~~[illegible]~~

ENO | PNO | DUR

Vertical Fragmentation

Cluster	Attributes
1	$\{A_4, A_5, A_7\}$
2	$\{A_4, A_5, A_6\}$

CA Matrix derived using the bond energy algorithm described on the book.