Hw1 soln

Sunday, 18 October 2020 22:01

2020280598

Yoke Kai Wen 叶凯雯 2020280598 HW1 : Distributed Database Design

Problem 5.2

Simple predicates $Pr = \{P_1: RESP='Manager', P_2: RESP='Analyst', P_3: RESP='Consultant', P_4: RESP='Engineer', P_5: RESP='Programmer', P_6: DUR<20, P_7: DUR \geq 20\}$

Form minterms:

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 ✗

Note: m_6, m_7, m_{10} are empty.

Form fragments:

ASG_1

ENO	PNO	RESP	DUR
E1	P1	Manager	12

ASG_2

ENO	PNO	RESP	DUR
E5	P2	Manager	24
E6	P4	Manager	48

ASG_3

ENO	PNO	RESP	DUR
E2	P2	Analyst	6

ASG_4

ENO	PNO	RESP	DUR
E2	P1	Analyst	24

ASG_5

ENO	PNO	RESP	DUR
E3	P3	Consultant	10

ASG_8

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

ASG_9

ENO	PNO	RESP	DUR
E4	P2	Programmer	18

Problem 5.8

Assume each query accesses attributes once per execution.

Access freq at diff sites: $q_1 \begin{bmatrix} 10 & 20 & 0 \\ 0 & 20 & 10 \end{bmatrix}$ (q_1: S_1, S_2, S_3; q_2: S_1, S_2, S_3)

Attribute usage matrix:

EMP: A1 = ENO, A2 = ENAME, A3 = TITLE
ASG: B1 = ENO, B2 = PNO, B3 = RESP, B4 = DUR

$q_1 \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$ (q_1: A_1, A_2, A_3; q_2: A_1, A_2, A_3)

$q_1 \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}$ (q_1: B_1, B_2, B_3, B_4; q_2: B_1, B_2, B_3, B_4)

Affinity matrix:

$\begin{matrix} & A_1 & A_2 & A_3 \\ A_1 & 30 & 30 & 30 \\ A_2 & 30 & 30 & 30 \\ A_3 & 30 & 30 & 30 \end{matrix}$ ← all attributes same affinity

$\begin{matrix} & B_1 & B_2 & B_3 & B_4 \\ B_1 & (60) & 30 & 30 & 60 \\ B_2 & 30 & 30 & 30 & 30 \\ B_3 & 30 & 30 & 30 & 30 \\ B_4 & 60 & 30 & 30 & (60) \end{matrix}$ → $B_1 \leftrightarrow B_4$ high affinity

Intuitively, EMP attributes should stay together in one fragment and be totally replicated in both S1, S2;

ASG attributes can be separated into {B1, B2, B4} and {B1, B2, B3} (B1 is primary key) + B2

Problem 5.17

Reject vertical fragmentation, because from previous part, EMP attributes have similar affinity and cannot be reasonably vertically fragmented, while fragmenting ASG vertically would mean storing extra colums since {ENO, PNO} are primary keys in ASG which outweighs the advantages of vert frag, therefore vert. frag does not make sense.

∴ Adopt horizontal fragmentation: For ASG, horiz frag based on DUR=24 predicate,
For EMP, horiz frag based on TITLE='Programmer' predicate.

Form fragments:

ASG

F_1

ENO	PNO	RESP	DUR
E2	P1	Analyst	24
E5	P2	Manager	24

F_2

ENO	PNO	RESP	DUR
E1	P1	Manager	12
E2	P2	Analyst	6
E3	P3	Consultant	10
E3	P4	Engineer	48
E4	P2	Programmer	18
E6	P4	Manager	48
E7	P3	Engineer	36
E8	P3	Manager	40

EMP

F_3

ENO	ENAME	TITLE
E4	J. Miller	Programmer

F_4

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anly

Allocation:

Sites	Fragments
S1	F_1, F_3, F_4
S2	F_1, F_2, F_3, F_4
S3	F_1, F_2

Justification

1) S2 is visited by both q_1, q_2 most frequently, thus it should contain all fragments accessed by q_1, q_2.
2) 60% of q_1 accesses are updates, 40% are reads. Usually we try to avoid replication when updates > reads, but in this case, the ratio is comparable. Furthermore, S1 ↔ S2 communication is slow, therefore we should have q_1 fragments at both sites as far as possible
3) Although F_4 is not accessed by q_1, q_2, it should still be stored for completeness.
4) S2 ↔ S3 communication rate is quite high, so it might be viable to store F_1, F_2 only at S2, and let q_3 located at S3 retrieve data from S2. But from reliability perspective, better to store copies at S3.