HE YIYANG 22100143D

QA.

(1) Block anchor can be seen as the first record in each block. It can ~~be seenbe used~~ be used to link the record block with the index block. The block anchor usually includes a key value or range that represents the minimum or maximum key of the data records stored within the block. When we search for a record, we can search the index block first and then block anchor will tell us the next datablock.

(2) The primary index is constructed on primary key of the relation table so only one primary index. The secondary index is used for non-key attributes. So Data block will not be affected when we applied those indices, so we can have many secondary index.

(3) Clustering index:

index file entry size = 4 + 6 = 10 bytes

index bfr = $\lfloor 600/10 \rfloor$ = 60 ~~entries/block~~ entries/block

\# index blocks = $\lceil 10/60 \rceil$ = 1 block access

\# of record (CID = 3) = 4000/10 = 400

blocking ~~factor~~ bfr = $\lfloor 600/100 \rfloor$ = 6 records/block

data blocks = $\lceil 400/6 \rceil$ = 67 blocks

\# block access = 1 + 67 = 68 blocks

Secondary index:

\# of record (DID = 16) = 4000/200 = 20 records/block

index file entry size = 4 + 6 = 10 bytes

index bfr = $\lfloor 600/10 \rfloor$ = 60 entries

\# index blocks = $\lceil 200/60 \rceil$ = 4 blocks

\# Record pointer block = $\lceil 20/100 \rceil$ = 1 block

\# block access = $\lceil \log_2 4 \rceil$ + 1 + 20 = 23 ~~blocks~~ blocks

Bitmap index:

bitmap size for one record = $|+| = 2$ bits

total space: $2 \times 4000 = 8000$ bits

$8000/8 = 1000$ bytes.

block for bitmaps: $\lceil 1000/600 \rceil = 2$ blocks

Searching cause 2 blocks access

records $= 4$

block access $= 4 + 2 = 6$ blocks

as $6 < 23 < 68$, bitmap is most efficient.

QB.

(1)

	A	B	C
1st tuple	2	Z	6
2nd tuple	7	X	8
3rd tuple	9	Y	2
4th tuple	8	Y	1

(2) ~~Ret~~ Relation R bfr $= \lfloor 1000/20 \rfloor = 50$ records / block
Number of block string $B_R = \lceil 100{,}000/50 \rceil = 2000$ blocks

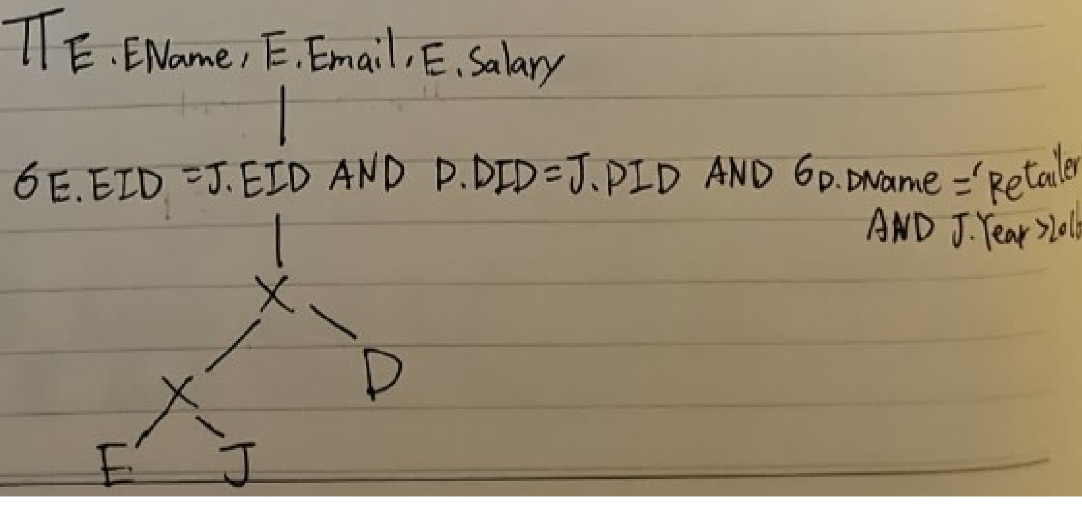
Relation S bfr $= \lfloor 1000/50 \rfloor = 20$ records / block
Number of block ~~access~~ string $B_S = \lceil 50000/20 \rceil = 2500$ blocks

Total number of block access $= B_R + B_R \times B_S = 2000 + 2000 \times 2500$
$= 5{,}002{,}000$ block access

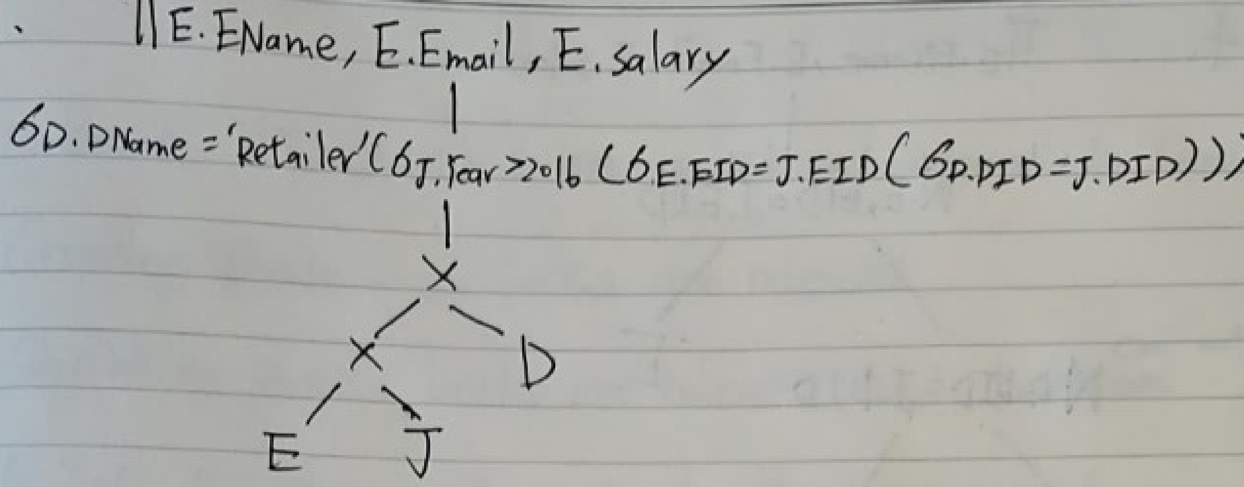
(3) Total number of block access $= B_S + B_R \times B_S = 5{,}002{,}500$ block access

Because $5{,}002{,}500 > 5{,}002{,}000$, So change join order can not improve the efficiency.

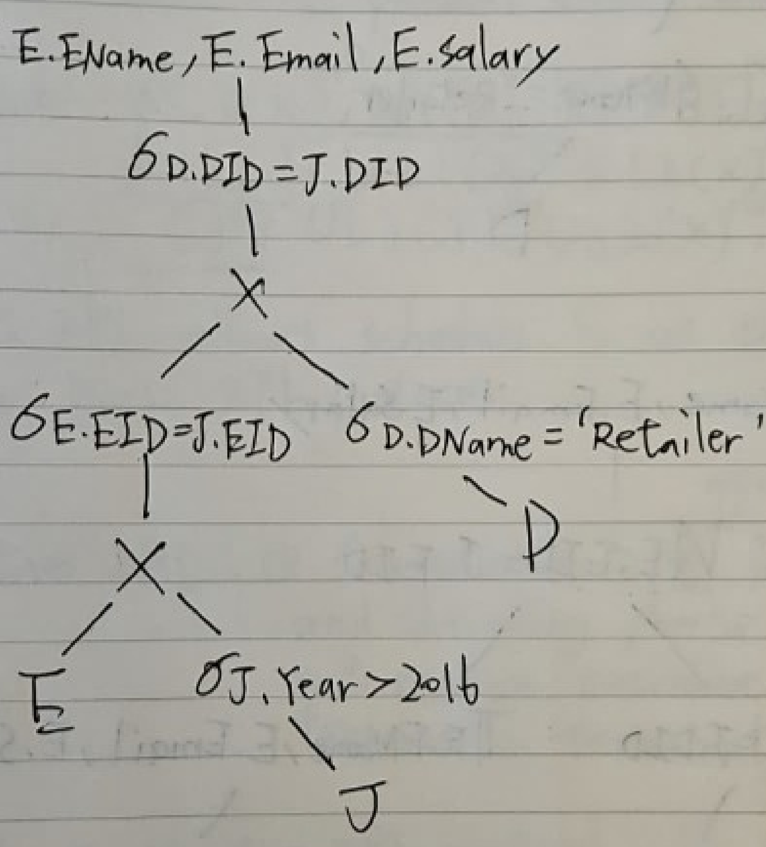
(4) Initial tree is:



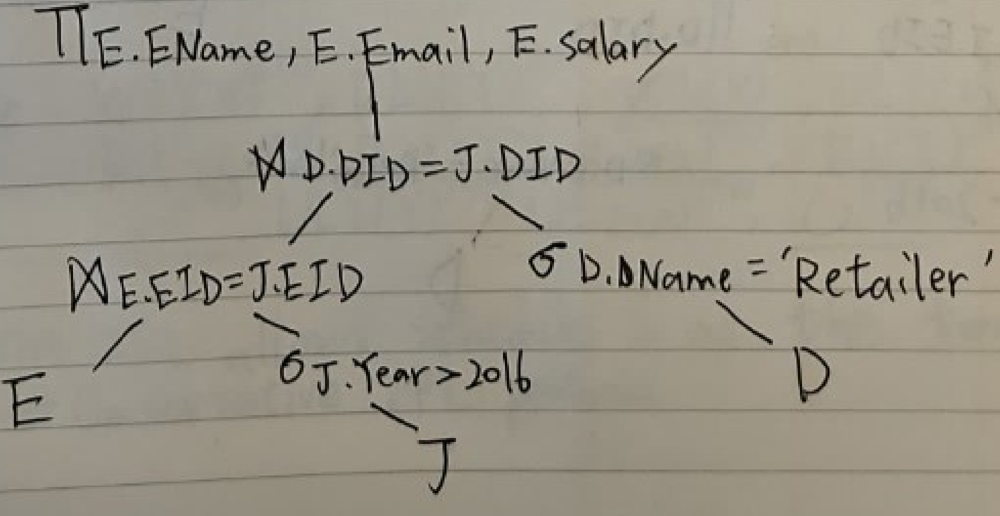
(5) 1.


ΠE.EName, E.Email, E.salary
σD.DName='Retailer'(σJ.Year>2016(σE.EID=J.EID(σD.DID=J.DID)))
×
×
D
E
J

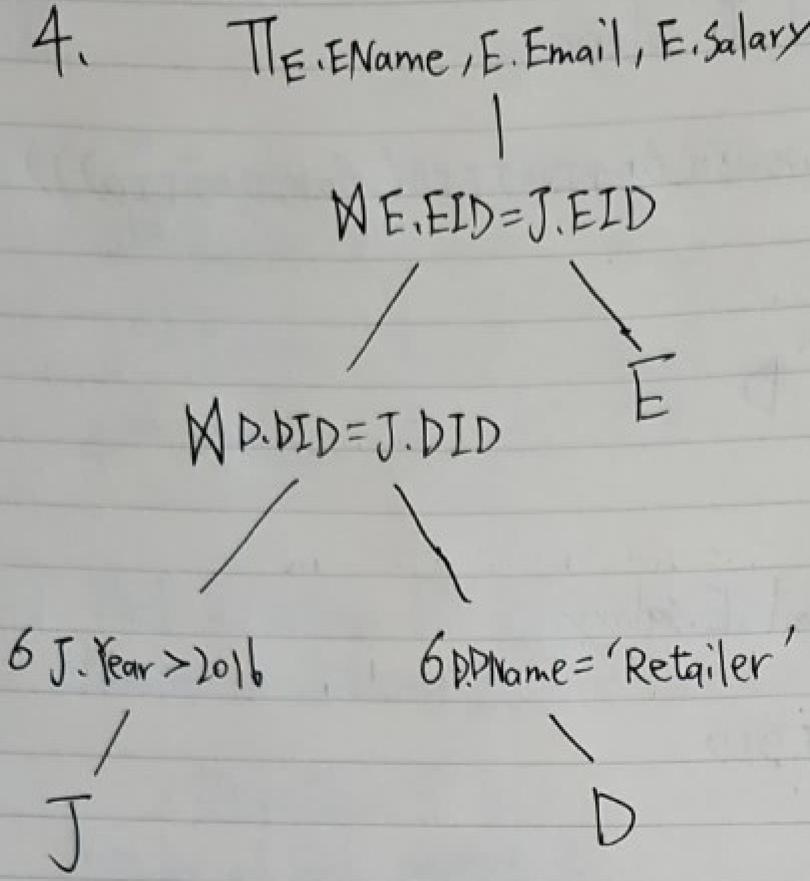
2.


ΠE.EName, E.Email, E.salary
σD.DID=J.DID
×
σE.EID=J.EID
σD.DName='Retailer'
×
D
E
σJ.Year>2016
J

3.

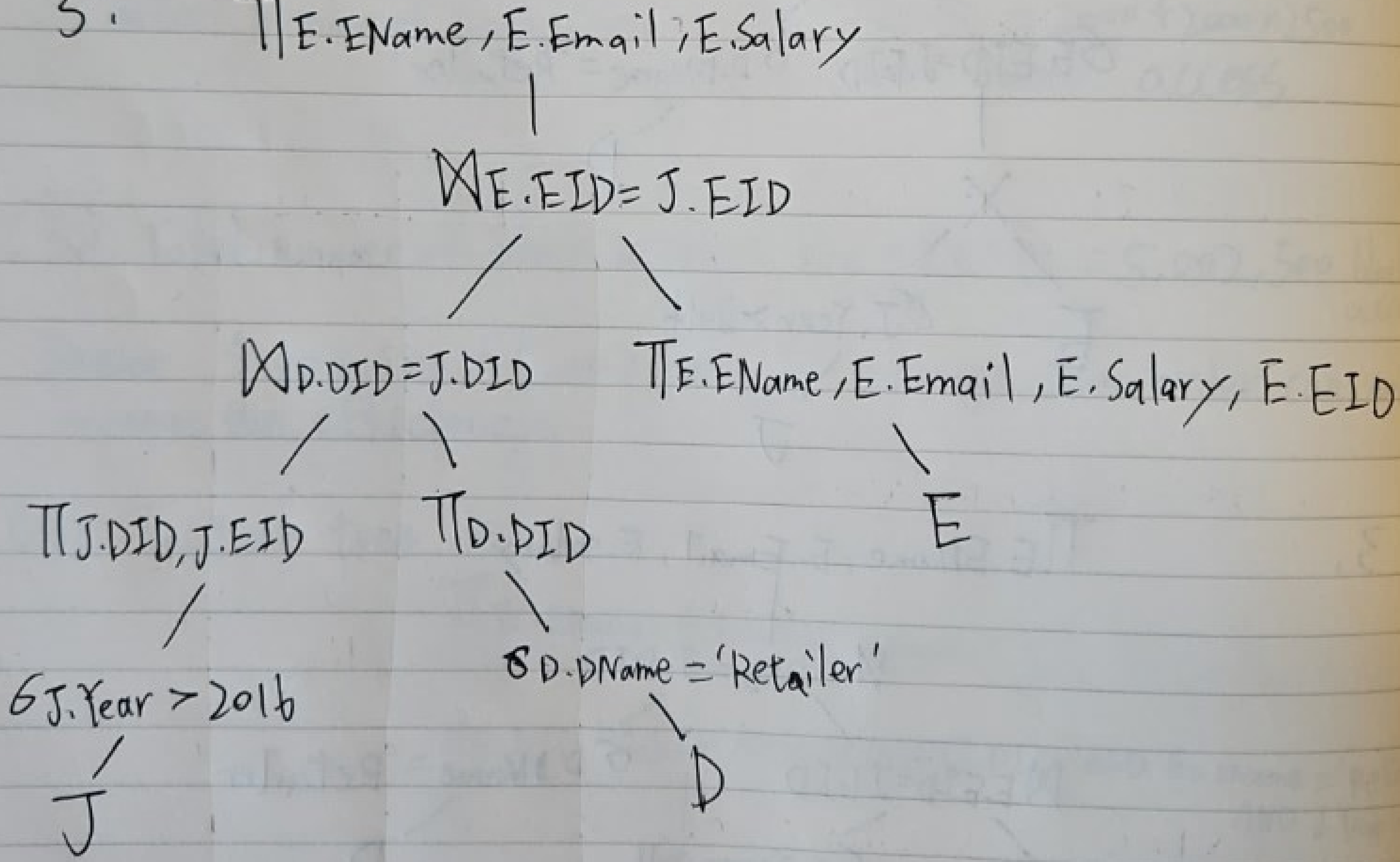

ΠE.EName, E.Email, E.salary
⋈D.DID=J.DID
⋈E.EID=J.EID
σD.DName='Retailer'
E
σJ.Year>2016
D
J

4.



∏E.EName, E.Email, E.Salary
⋈E.EID=J.EID
E
⋈D.DID=J.DID
σJ.Year>2016
σD.DName='Retailer'
J
D

5.



∏E.EName, E.Email, E.Salary
⋈E.EID=J.EID
⋈D.DID=J.DID
∏E.EName, E.Email, E.Salary, E.EID
E
∏J.DID, J.EID
∏D.DID
σD.DName='Retailer'
σJ.Year>2016
J
D

QC.

(1) Basic 2PL: ① Locks are applied and removed in two phases:

Crowing phase: no locks are released

Shrinking phase: Locks are released and locks ~~are~~ can not be acquired.

② Yes, the Basic 2PL protocol allows schedule S.

③ Schedule: WL1(X)$^{(T1)}$, W1(X)$^{(T1)}$, RL2(Y)$^{(T2)}$, R2(Y)$^{(T2)}$, RL1(Y)$^{(T1)}$, R1(Y)$^{(T1)}$, UL1(X)$^{(T1)}$, RL2(X)$^{(T2)}$, R2(X)$^{(T2)}$, UL1(Y)$^{(T1)}$, C1$^{(T1)}$, UL2(Y)$^{(T2)}$, UL2(X)$^{(T2)}$, C2$^{(T2)}$

Basic 2PL allows schedule S as the two transactions follow Basic 2PL protocal

Conservative 2PL: ① Conservation 2PL also follows crowing phase and shrinking phase. However, it will not release exclusive locks until the transaction terminates.

② Yes, the Conservative 2PL protocal allows schedule S

③ Schedule: WL1(X)$^{(T1)}$, RL1(Y)$^{(T1)}$, W1(X)$^{(T1)}$, UL1(X)$^{(T1)}$, RL2(Y)$^{(T2)}$, RL2(X)$^{(T2)}$, R2(Y)$^{(T2)}$, R2(X)$^{(T2)}$, R1(Y)$^{(T1)}$, UL1(Y)$^{(T1)}$, C1$^{(T1)}$, UL2(Y)$^{(T2)}$, UL2(X)$^{(T2)}$, C2$^{(T2)}$

Conservative 2PL allows sehedule S as the two transactions follow ~~#~~ Conservative 2PL protocal.

Strict 2PL: ① The strict 2PL also follow crowing phase and shrinking phase but all exclusive (write) locks (held by transaction) released until transaction stop.

② No, the strict 2PL protocol does not allows shedule S

③ after WL1(X)^(T1) the transaction shoulld continue to finish So RL2(X) and R2(X) can not held before C1

So schedule S: W1(X), R2(Y), R1(Y), R2(X), C1, C2

Can not happen

So strict 2PL does not allow schedule S.

(2)

strict 2PL

according to the schedual T_1 acquires exclusive lock on X before T2.

The exeation of T2 need to wait T1 unlock X.

T1	T2
Writ-lock1(X)	
R1(X)	
W1(X)	
Write_lock1(Y)	
R1(Y)	
W1(Y)	
UnLock1(X)	
UnLock1(Y)	
A1	
	Write_Lock2(X)
	R2(X)
	W2(X)
	UnLock2(X)
	C2

T_1 can only unlock X when it aborts After T_1 aborted, X back to its value.

T_2 will not read the dirty version of X ~~and~~ So no dirty read problem