1. It is not BCNF; We can first move $F \to C$ to its own table, i.e. $R_1(A, B, D, E, F)$ and $R_2(F, C)$. Then we decompose R_1 into $R_3(A, B, D, E)$ and $R_4(B, D, F)$ since $BD \to F$. Therefore the final set of tables is $R_2(F, C), R_3(A, B, D, E), R_4(B, D, F)$.

2. (a) We have $\{C, D, E, F, G, H\}$ that has no dependencies, which makes $2^6 = 64$ closed sets. Then since a closed set with B must also have C, D in it, we have another $2^4 = 16$ sets (since $\{E, F, G, H\}$ makes 16 combinations). Then we can add A to those 16 sets to make 16 more. Therefore we have $64+16+16 = 96$ closed sets in total.

   (b) Any subset that implies $A \to B$ must contain either or both $\{A \to B\}$ and $\{A \to C, C \to B\}$. Then we can make 8 combinations that does not logically imply $A \to B$ with $\{B \to A, C \to A, B \to C\}$, then another 8 from those combinations with $A \to C$ added (and not $C \to B$), simiarly, we have another 8 with $C \to B$ added. Therefore the number of subsets that implies $A \to B$ is $64 - 8 - 8 - 8 = 40$.

3. I: CBAD: guaranteed to empty; Table C can be deleted first since no other table depends on it. Then B can be deleted since although D(z1) depends on B(x) it has ON DELETE SET NULL specified and it is nullable. A can now be deleted since D(z2) is also nullable.

   II: CDAB: not guaranteed to empty; A is deleted before B so B(x) will be set NULL but it is a PRIMARY KEY and cannot be set NULL.

   III: BCDA: guaranteed to empty; B is deleted first and D(z1) is set NULL, C is then deleted since nothing references it. D can then be deleted since A is not dependent on it.

4. (a)

	INSERT	DELETE	UPDATE
Executive	YES (insert high pay)	NO	YES (raise pay past 10 times)
Employee	YES (insert new lowest pay)	NO	YES (lower lowest pay)

   (b) CHECK (salary <= (SELECT 10*MIN(salary) FROM Employee WHERE division = div_in_charge))

5. SELECT publisher, SUM(price) as a1
   FROM Books
   WHERE birthYear>1950
   GROUP BY publisher
   HAVING SUM(price)>10000;

6. (a) missing the average rotational latency - r, since the head might not start at the beginning of the file; Access time = seek time + rotational delay + transfer rate = 10ms + r + 1/(6000/60) = 10ms + r + 10ms = r + 20ms.

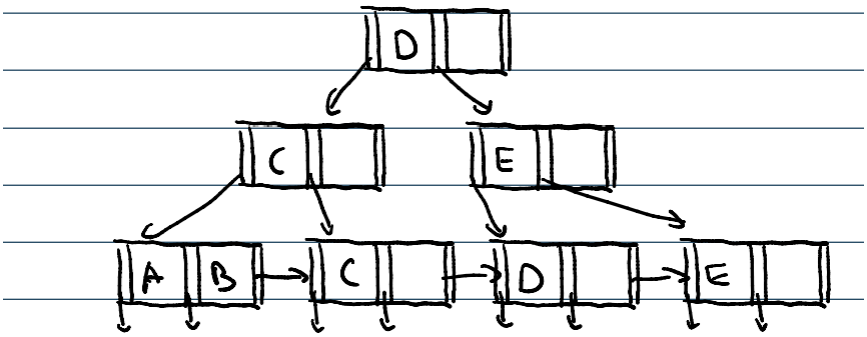
   (b) 9804; $\lfloor \frac{\text{4096 bytes/block}}{\text{40 bytes/tuple}} \rfloor = 102$ tuples/block, $\lceil \frac{\text{1000000 tuples/table}}{\text{102 tuples/block}} \rceil = 9804$ blocks/table.

   (c) Each block is 4096 bytes so each one can fit 255 longs and 256 pointers. So we have $\lceil \frac{1000000}{255} \rceil = 3922$ leaf nodes, $\lceil \frac{3922}{256} \rceil = 15$ level 2 nodes, and $\lceil \frac{15}{256} \rceil = 1$ root node. Therefore we need at least 3922+15+1 = 3938 ndoes.

   (d) 3 IOs; 2 IOs from root to level 3, 1 IO from level 3 to data.

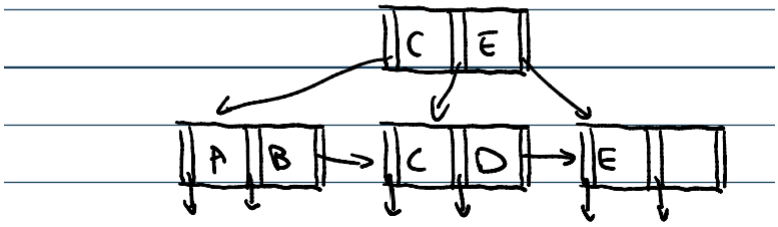
7. (a)

Height-3 Tree:

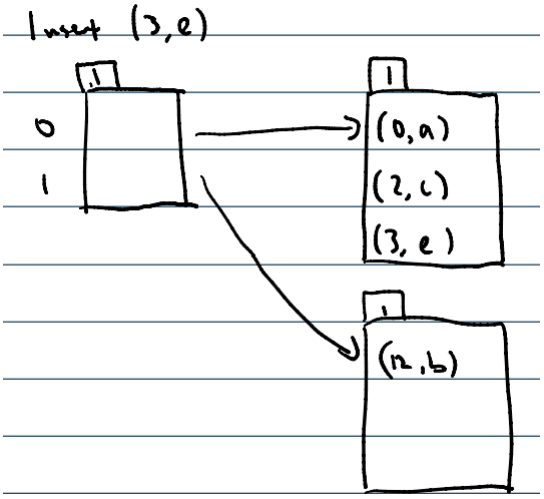


(b)

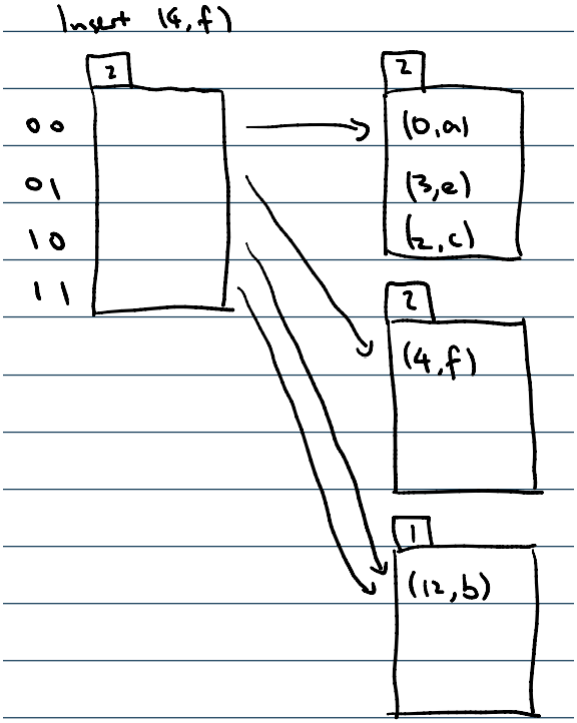
Height-2 Tree:



8. (a)



(b)


insert (4, f)
2
00
01
10
11
2
(0, a)
(3, e)
(2, c)
2
(4, f)
1
(12, b)