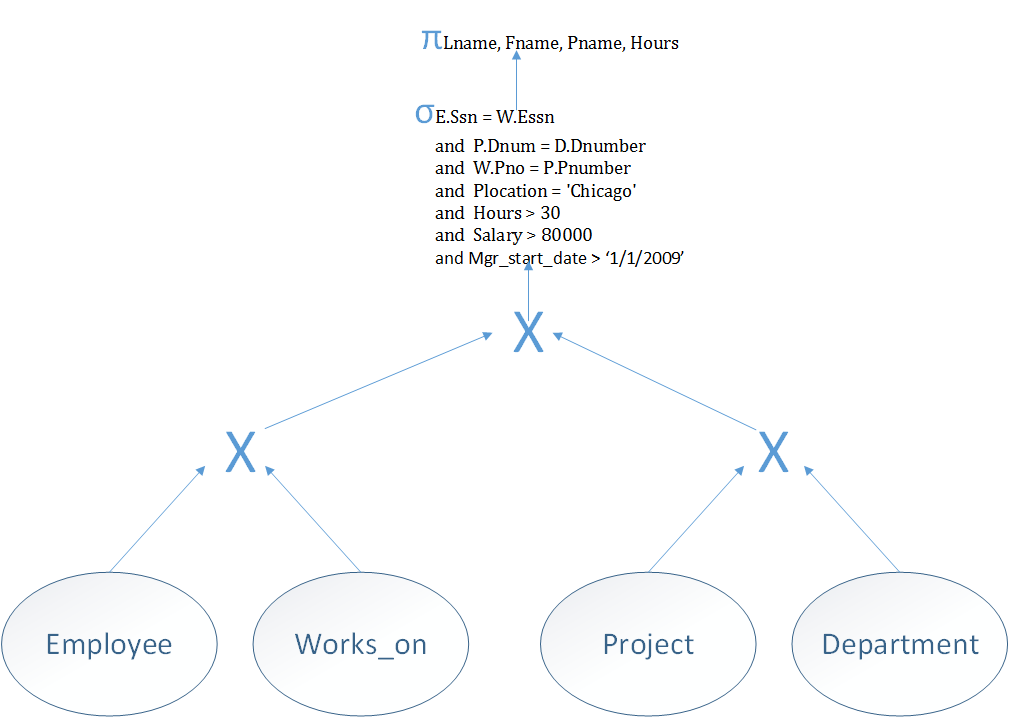
* 1. Record length R = (30+9+9+40+9+8+1+4+4) + 1 = 115 bytes
  2. bfr=floor(B/R) = floor(512/115) = 4 records per block  
     No of blocks needed for file = ceiling(r/bfr) = ceiling(30000/4) = 7500 blocks  
     1. Index record size Ri = (Vssn + P) = (9 + 6) = 15 bytes  
        Index blocking factor bfri = fo = floor(B/Ri) = floor(512/15) = 34
     2. No of first level index entries r1 = no of file blocks b = 7500 entries  
        No of first level index blocks b1 = ceiling(r1/bfri) = ceiling(7500/34) = 221 blocks
     3. No of second level index entries r2 = no of first level blocks b1 = 221 entries  
        No of second level index blocks b2 = ceiling(r2/bfri) = ceiling(221/34) = 7 blocks  
        r3 = b2 = 7 entries  
        b3 = ceiling(r3/bfri) = ceiling(7/34) = 1  
        Since the third level has only one block, it is the top index level. Therefore, there are 3 index levels.
     4. Total no of blocks bi = b1+b2+b3 = 221+7+1 = 229 blocks
     5. No of block accesses to search for a record = x+1 = 3+1 = 4.
     6. Ri = (Vssn + P) = (9+6) = 15 bytes  
        bfri = fo = floor(B/Ri) = floor(512/15) = 34 index records per block
     7. R1 = no of file records r = 30000  
        b1 = ceiling(r1/bfri) = ceiling(30000/34) = 883 blocks
     8. R2 = b1 = 883 entries  
        b2 = ceiling(r2/bfri) = ceiling(883/34) = 26 blocks  
        r3 = b2 = 26 entries  
        b3 = ceiling(r3/bfri) = ceiling(26/34) = 1 block  
        Hence, the index has 3 levels.
     9. Total no of blocks bi = b1+b2+b3 = 883+26+1 = 910 blocks
     10. No of block accesses to search for a record = x+1 = 4.
     11. Ri = (Vdepartmentcode + P) = (9+6) = 15 bytes  
         bfri = floor(B/Ri) = floor(512/15) = 34 index blocks
     12. Avg no of records for each value = r/1000 = 30000/1000 = 30  
         Since Pr = 7 bytes, the number of bytes needed at the level of indirection for each value of DEPARTMENTCODE is 7\*30 = 210 bytes, which fits in one block. Hence, 1000 blocks are needed for the level of indirection.
     13. R1 = no of distinct values of DEPARTMENTCODE = 1000 entries  
         B1 = ceiling(r1/bfri) = ceiling(1000/34) = 30 blocks
     14. R2 = b1 = 30 entries  
         B2 = ceiling(r2/bfri) = ceiling(30/34) = 1  
         Hence, he index has x=2 levels
     15. Total no of blocks = bi = b1+b2+b indirection = 30+1+1000 = 1031 blocks
     16. No of block accesses = x+1 = 2+1 = 3 block accesses   
         If we assume that the 30 records are distributed over 30 distinct blocks, we need an additional 30 block accesses to retrieve all 30 records. Hence, total block accesses needed on average to retrieve all the records with a given value for DEPARTMENTCODE = x+1+30 = 33.
     17. Ri = (Vdepartmentcode + P) = 9+6 = 15 bytes  
         bfri = fo = floor(B/Ri) = floor(512/15) = 34 index records per block
     18. R1 = no of distinct DEPARTMENTCODE values = 1000 entries  
         b1 = ceiling(r1/bfri) = ceiling(1000/34) = 30 blocks
     19. R2 = b1 = 30 entries  
         b2 = ceiling(r2/bfri) = ceiling(30/34) = 1  
         Hence, index has x=2 levels.
     20. Total no of blocks bi = b1+b2 = 30+1 = 31 blocks
     21. No of block accesses to search for the first block in the cluster of blocks = x+1=3.   
         The 30 records are clustered in ceiling(30/bfr) = ceiling(30/4) = 8 blocks.   
         Therefore, total block accesses needed on average to retrieve all the records with a given DEPARTMENTCODE = x+8 = 2+8 = 10 block accesses.
     22. For a B + -tree of order p, the following inequality must be satisfied for each internal tree node: (p \* P) + ((p - 1) \* V SSN ) < B, or (p \* 6) + ((p - 1) \* 9) < 512, which gives 15p < 521, so p=34. For leaf nodes, assuming that record pointers are included in the leaf nodes, the following inequality must be satisfied: (p leaf \* (V SSN +P R )) + P < B, or (p leaf \* (9+7)) + 6 < 512, which gives 16p leaf < 506, so p leaf =31.
     23. Assuming that nodes are 69% full on the average, the average number of key values in a leaf node is 0.69\*p leaf = 0.69\*31 = 21.39. If we round this up for convenience, we get 22 key values (and 22 record pointers) per leaf node. Since the file has 30000 records and hence 30000 values of SSN, the number of leaf-level nodes (blocks) needed is b 1 = ceiling(30000/22) = 1364 blocks.
     24. We can calculate the number of levels as follows: The average fan-out for the internal nodes (rounded up for convenience) is fo = ceiling(0.69\*p) = ceiling(0.69\*34) = ceiling(23.46) = 24 number of second-level tree blocks b 2 = ceiling(b 1 /fo) = ceiling(1364/24) = 57 blocks number of third-level tree blocks b 3 = ceiling(b 2 /fo) = ceiling(57/24)= 3 number of fourth-level tree blocks b 4 = ceiling(b 3 /fo) = ceiling(3/24) = 1 Since the fourth level has only one block, the tree has x = 4 levels (counting the leaf level).
     25. Total no of blocks for the tree bi = b1+b2+b3+b4 = 1364+57+3+1 = 1425 blocks
     26. No of block accesses to search for a record = x+1 = 4+1 = 5.

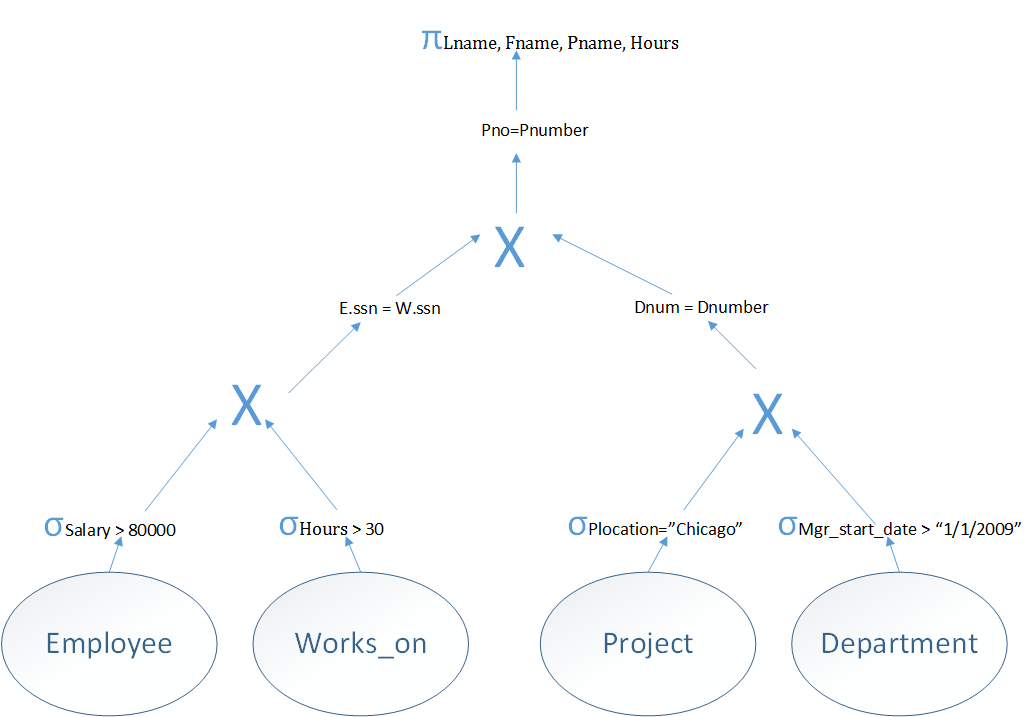
1. The Query tree for the above tree can be written as shown below:



Applying Heuristic rules and optimizing it further:

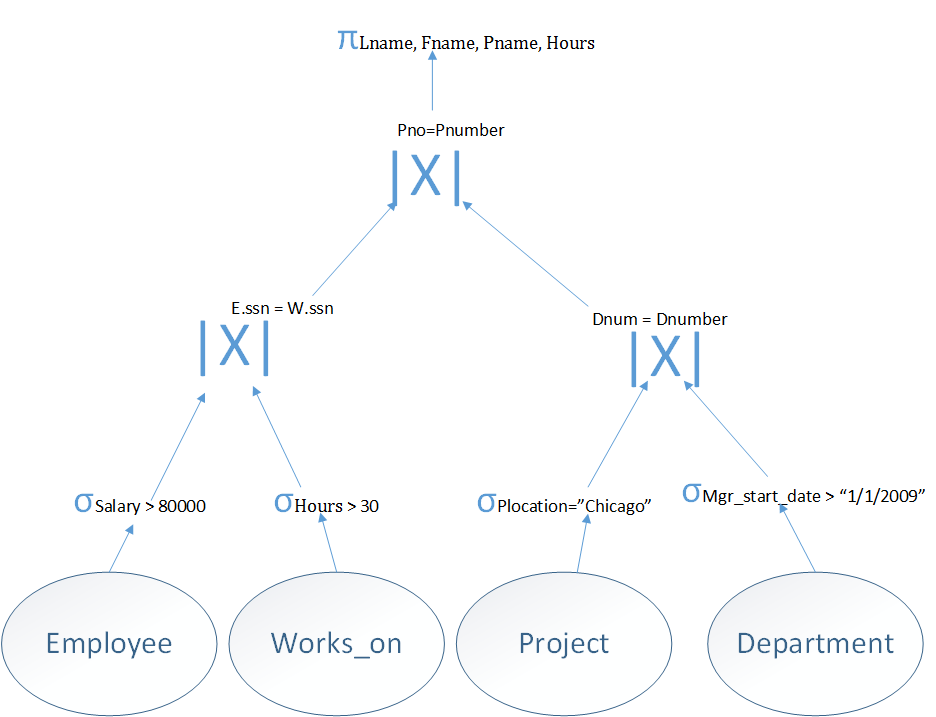
Rule 1:

Conjunctive selections are broken up into separate individual selections



Rule 2:

Combining Cartesian product and selection to form Join operation



Rule 3:

Conjunctive projections are broken up into required projections on each level and reordering.

