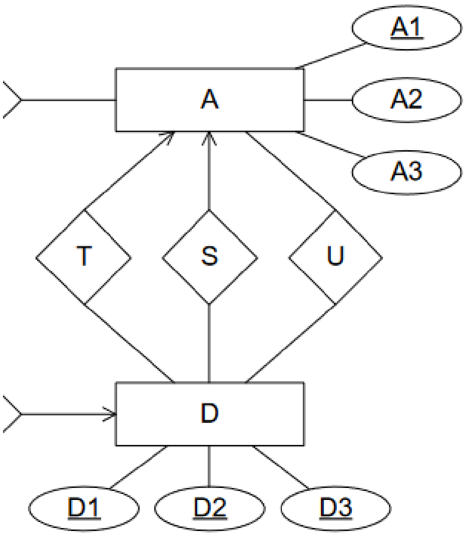
Accumulated Questions

1. Should we use cardinality in Ternary relationships?
2. When we design the ER diagram should be include notes explaining different solutions and why this one we choose is the better one?
3. Some notes for the er diagrams on the exercise book are not exactly right regarding the constrain. More precisely the ones about making sure the DIRECTOR works at a hospital. (You can indirectly find more information about the director which can be used as a constrain). So my question is can indirect concluded information be used as a constrain in ER diagrams? I know it is not very precise question but it is really hard to not explain it in person. (You can skip this)
4. Are there cases where we can not model the requirements in ER diagrams exactly (including all the constrains)? If not what should we do (explain in notes, ...)?
5. How to represent several foreign keys of the same element? Example:

   $D(\underline{D1}, \underline{D2}, \underline{D3}, \underline{\underline{TA1}}, \underline{\underline{SA1}}, \underline{\underline{SA2}})$ I think this is right. (You can skip)
6. For exercise 6) b (exercise book) for the ER at 3, you don't need the DEPT_STAFF at all? Why is it included?
7. How do you calculate the size of a hash table:
   - we never took into account the pointers between the blocks (not the ones of the hash table)
   - do we count this between blocks pointers in true size or in used disk size?
   - should a block fit some records and one pointer at the end.
   - Average bucket size: should it be a rounded number or a decimal. (I think that averages should be decimals. If this is true then Janos may have made some small mistakes)
8. Do we take into account the hash table pointers between blocks in a bucket? In practice we somewhere did and somewhere didn't so it is very confusing.
9. To calculate the layers of the tree do you need + 1 for the number of blocks before you take the log of br?
10. 36 (book exercises) in calculation you used the pointer size as 8bytes but in the exercise it says it is 64bytes. Maybe it was supposed to be bits.

11. In practice we had that the cost of the join is $C_{join} = b_r + n_r * C_s$ (r(r) and s(S)) but the problem is that we are supposing that n_r only needs to read one block but we should check (if not ordered) every block $C_{join} = b_r + n_r * b_s$. If ordered than the for each n_r you only needs several blocks. If the matching record fit in one block => $C_{join} = b_r + n_r * C_s$ works but if not when it depends in the index if it supports interval reads. In the problem set we had the hash join but I don't think the solution we gave is the right one. We used $C_{join} = b_r + n_r * C_s$ but maybe the matching rows are scattered when you cant just access 1 block. Can you explain what exactly we should do? (Thinking more deeply you can hash using the column you are using for join => that the same values land of the column land in the same bucket so you kind of get an ordering but still you don't know if all the matching rows fit in one block)