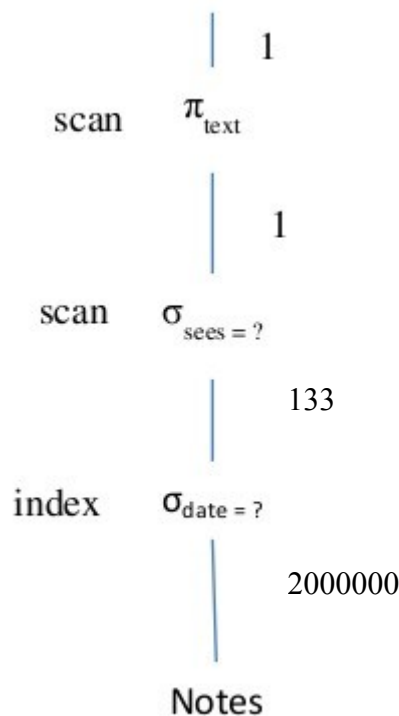1. There is a single table with two selection clauses for sees = ? and date =?. There is an index on the table for date and picking a single date is most likely highly selective due to the wide range of dates, so we can perform an indexed search on the 2000000 records and 15000 meeting days, so on an average there is (2000000/15000) = 133 meeting notes per date. Since there are 100 counselors, notes per counselor is almost 1. We then do a projection on text which must be a scan.

PS. The index on date is a secondary index, so the 133 matching records will each require a random I/O. If the Notes records were small, then it would be faster to do a scan. In fact, they are large, so the index is better. This is one of the few places where the record size makes a difference.

$$\text{scan} \quad \pi_{text} \quad \Big|\ 1$$

$$\Big|\ 1$$

$$\text{scan} \quad \sigma_{sees\ =\ ?}$$

$$\Big|\ 133$$

$$\text{index} \quad \sigma_{date\ =\ ?}$$

$$\Big|\ 2000000$$

Notes

2. The only table with a selection is the Client table with a selection on name. This selection is highly selective since names are nearly unique, and we can perform an indexed search to receive 1 result out of the 5000 clients. There is no other table with fewer entries than this so we start with the Client table. We can then join with the meeting table using an index nested loop since we expect to find approximately 2 (10000/5000) results for each client which is highly selective (2/10000 = .02%). We can then join with either Counselor or MeetingDay, however Counselor will produce 2 results since there is 1 Counselor per Meeting while MeetingDay will produce 3 results since there is approximately 1.5 MeetingDays per Meeting (15000/10000) so we will first join with the Counselor table. The join with Counselor will use a merge-join since the results are not highly selective (2/100 = 2%). The left join with MeetingDay will be done using an index-nested loop since the results are highly selective (3/15000 = 0.02%). There is then a projection on title and day which must be done using a scan.
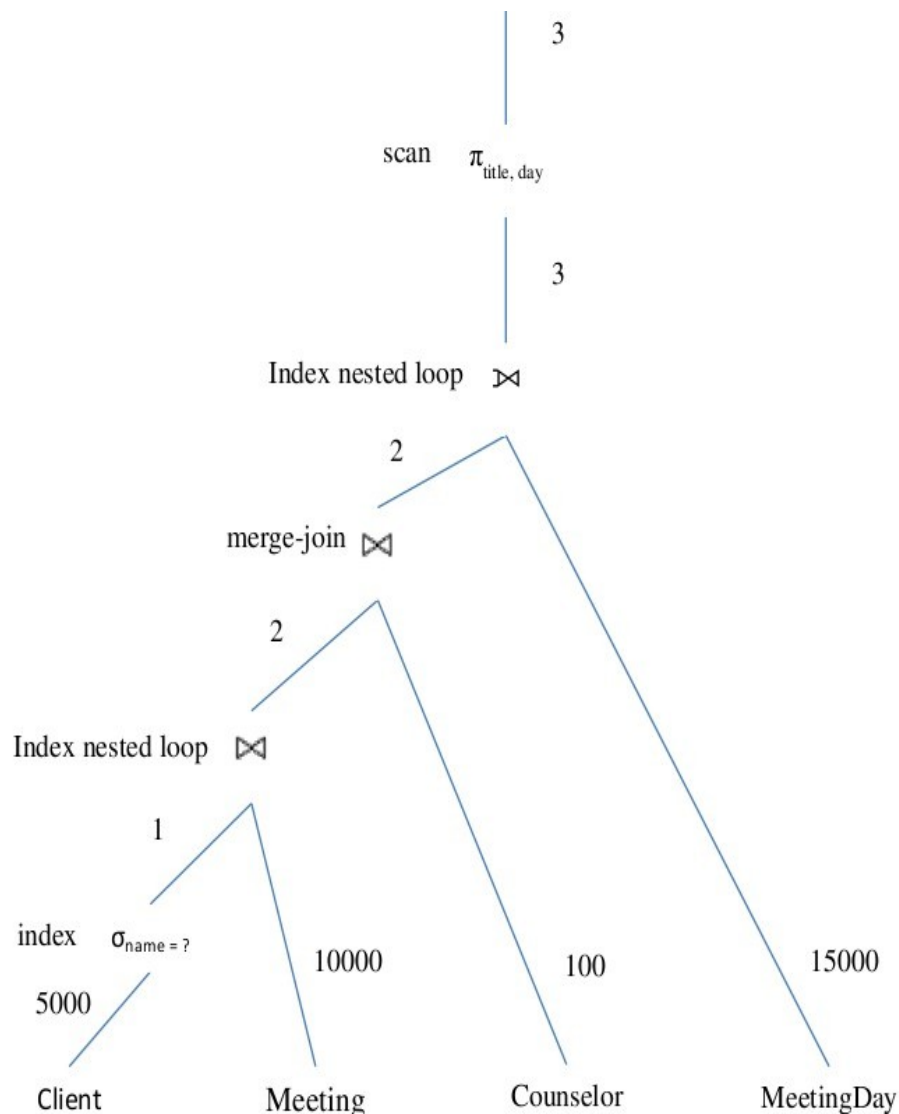
3. The table with the smallest selection is the Counselor table with a selection on title. There is no index on title so we must do a scan on the table and since titles are nearly unique we can expect to find a single result. There is no other table with fewer entries than this so we start with the Counselor table. Now, we can do a selection on MeetingDay table on name, this would give 2143 (15000 MeetingDay/7 days) results. We can then join the selection on Counselor and MeetingDay table. We then join with the Client table which can be done using an index nested loop since the results are highly selective (22/5000 = .0044 = 0.44%). The projection on name must be done using a scan, and then an order by must be done using a tree since order is needed.
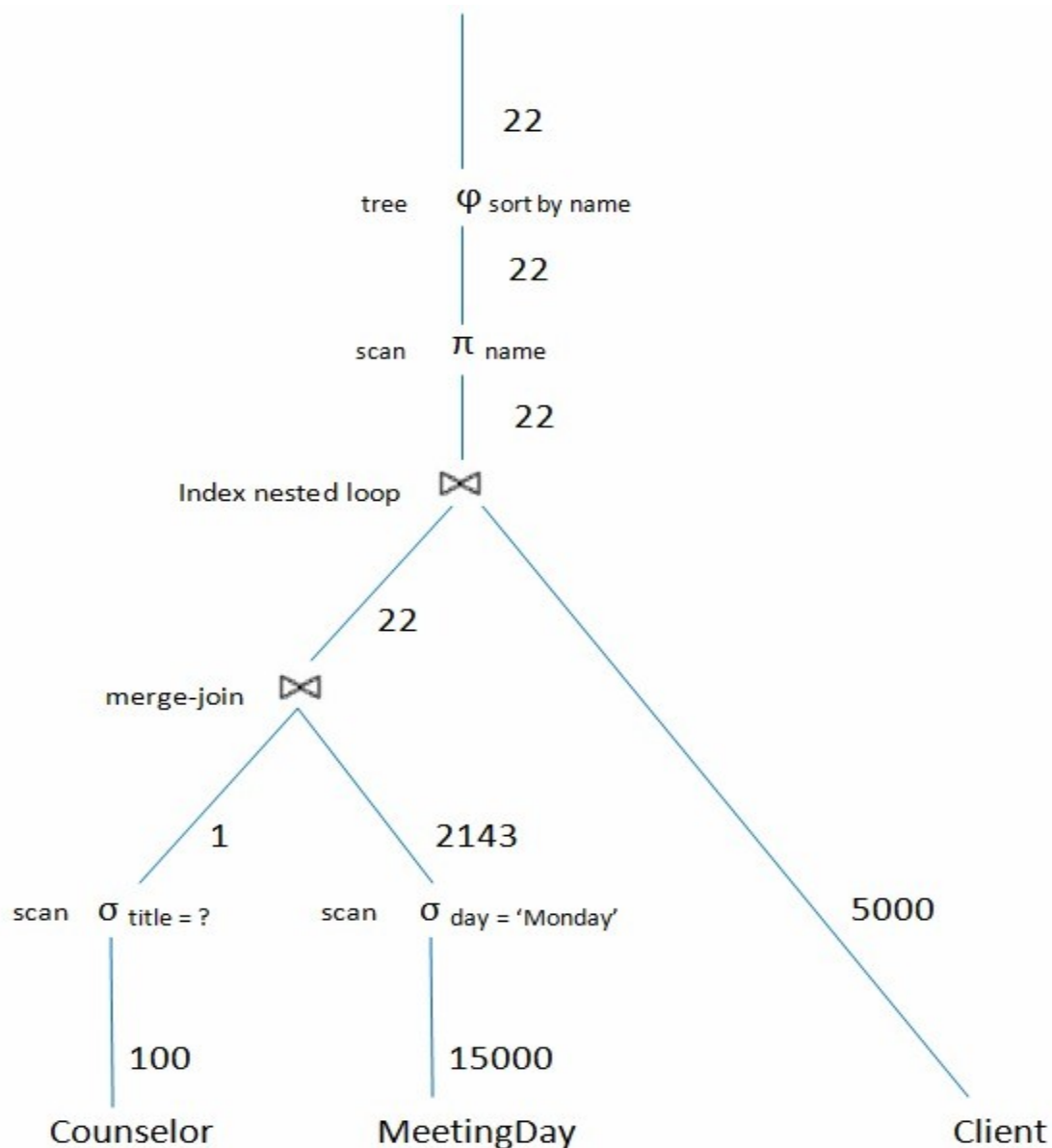
4. The only table with a selection is the Client table on name like 'P%'. The selection must be done using a scan since, assuming the start of names are evenly distributed (1/26), we are searching for a large range that is not highly selective. Although the Counselor table has fewer entries than this, if we start with the Counselor table and merge into Meeting we will result in 10000 entries, whereas using our client table with its selection results in only 386 entries. For this reason we start with the Client table and we can then join with the Meeting table, and since each client has approximately 2 meetings (10000/5000), we will use a merge-join since this is not highly selective. We next reach the outer query, where we join with the Counselor table using a right outer join which again is performed using a sort-merge since we will be going through each Counselor. We can assume that each counselor has 100 meeting (10000/100), and there is a 3.86% (386/10000) chance each counselor has a client whose name starts with P, so 96 counselors will get null entries in this join in addition to the 386 entries (96+396 = 482). We select Counselors that did not have a Client whose name started with P by selecting entries where Meeting's sees value is not null by scanning through the results, and we decided before that 96 counselors met this requirement. We then perform a projection on title (query in assignment had a typo and said to select co.name, although counselor's have a title instead of a name) which must also be done using a scan.
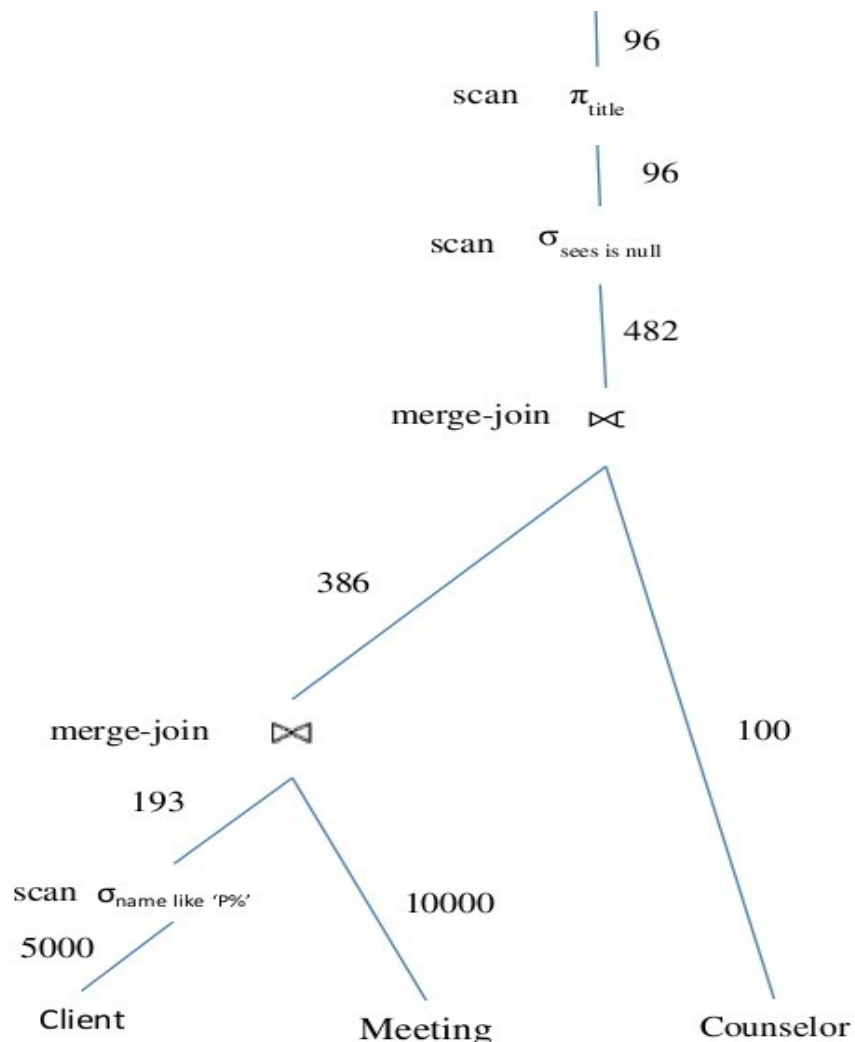
5. The only table with a selection is the Counselor table with a selection on title. The selection must be done using scan since we are searching for a larger range that is not highly selective. We will estimate that 10% of counselors are weight loss counselors for the sake of this query. We can then left join with the Meeting table for each Counselor to ensure that counselors that do not have meetings will be included. Each counselor has an average of 100 (10000/100) meetings, so we estimate there will be approximately 10*100 = 1000 records. This join will be done using a merge join since there is no index for counselors in Meeting. We can then right join this with the client table using a merge-join to find all the pairings of client and counselor with their connecting meetings. Each client has an average of 2 (10000/5000) meetings, so we estimate there will be approximately 2*5000 = 10000 records. We select clients where there is not a null entry for the counselor he or she has a meeting with. We then perform a projection on name which must be done using a scan.

$$5$$

scan $\quad \pi_{name}$

$$5$$

scan $\quad \sigma_{sees\ is\ null}$

$$10000$$

merge-join $\quad \bowtie$

$$1000$$

merge-join $\quad \bowtie$

$$10$$

$$5000$$

scan $\quad \sigma_{title\ like\ 'Weight\ Loss\ \%'}$

$$100$$

$$10000$$

Counselor $\qquad$ Meeting $\qquad$ Client