CS 448 Spring 2024 Homework 3

Sean Rettig

March 3, 2024

1 (1.0 Point)

Specify the following queries in relational algebra (R.A.) on the COMPANY database schema shown in Figure 5.5 in the Textbook. Do not use any other information from the database state on Figure 5.6 to build the queries.

(a) List names of employees (Fname, Minit, Lname) who are directly supervised by a female supervisor.

(b) List names of employees (Fname, Minit, Lname) who are NOT assigned to every project.

(c) List names of employees (Fname, Minit, Lname) in Departments with locations in Houston or Sugarland.

(d) List the names of department managers (Fname, Minit, Lname) who have no dependents.

Answer:

(a) $\pi_{\text{Fname, Minit, Lname}}(\text{EMPLOYEE} * (\rho_{(\text{Super_ssn})}(\pi_{\text{Ssn}}(\sigma_{\text{Sex=female}}(\text{EMPLOYEE})))))$

(b) (EMPLOYEE - (EMPLOYEE $\div \pi_{\text{Pnumber}}$(PROJECT)))

(c) $\pi_{\text{Fname, Minit, Lname}}($
EMPLOYEE $\bowtie_{\text{Dno=Dnumber}}$ ($\sigma_{\text{Dlocation='Houston' OR Dlocation='Sugarland'}}$(DEPT_LOCATIONS))
)

(d) $\pi_{\text{Fname, Minit, Lname}}($
(EMPLOYEE $\bowtie_{\text{Ssn=Mgr_ssn}}$ (DEPARTMENT))$*(\pi_{\text{Ssn}}(\text{EMPLOYEE}) - \rho_{(\text{Ssn})}(\pi_{\text{Essn}}(\text{DEPENDENT})))$
)

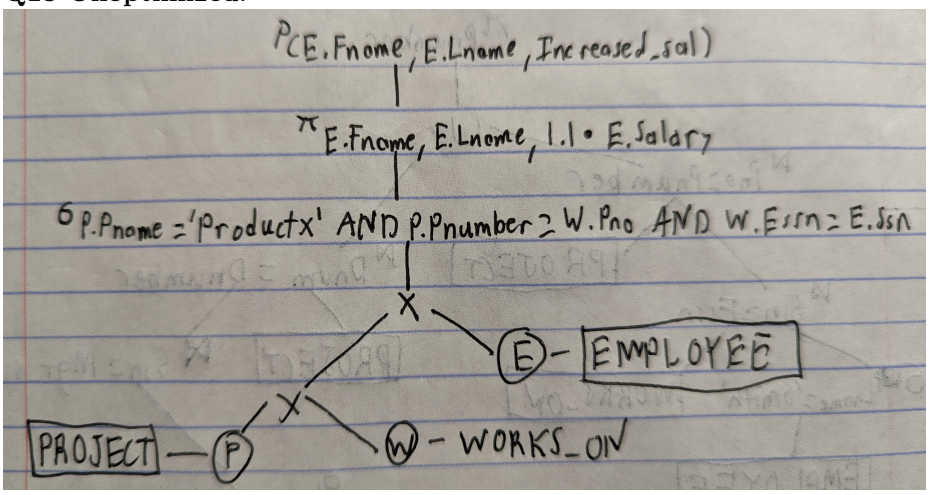
2 1.0 Point

Consider SQL queries Q13 and Q4A in Chapter 6.

(a) Draw the initial query tree for each of these queries, and then show how the query tree is optimized by the algorithm outlined in Section 19.1.

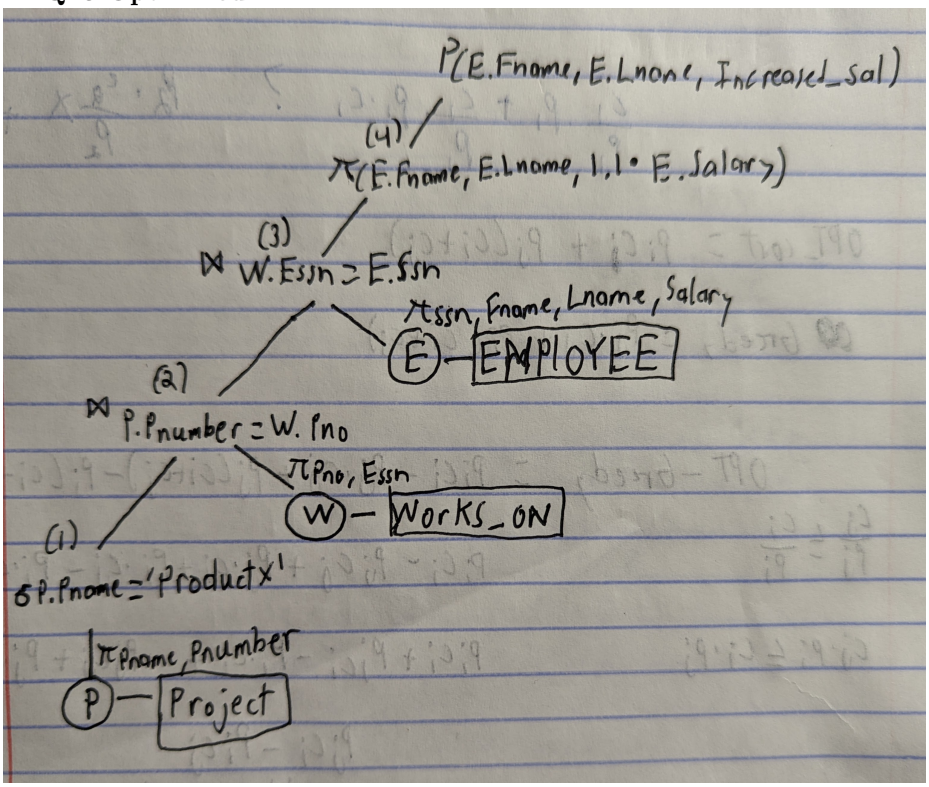
(b) For each query, compare the initial query and the final query tree. Discuss the difference between each tree.

Answer:
Q13 Unoptimized:

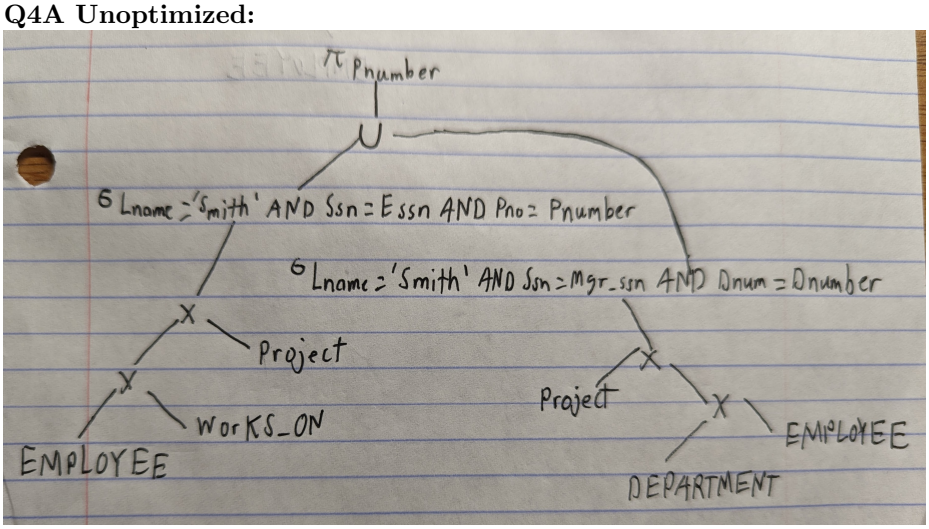


Step 1 specified a break up of any SELECT operations, and that was used to break up the third to final SELECT operation into multiple cascading SELECT Operations. I used Step 2 and 4 to move the SELECT Operations down as far as they can, as well as combining them with their respective CARTESION PRODUCT into a JOIN operation. I followed Step 3 by moving the most restricive SELECT down to the bottom of the tree. Step 5 was used to move project operations as far down the tree as needed.

Q13 Optimized:

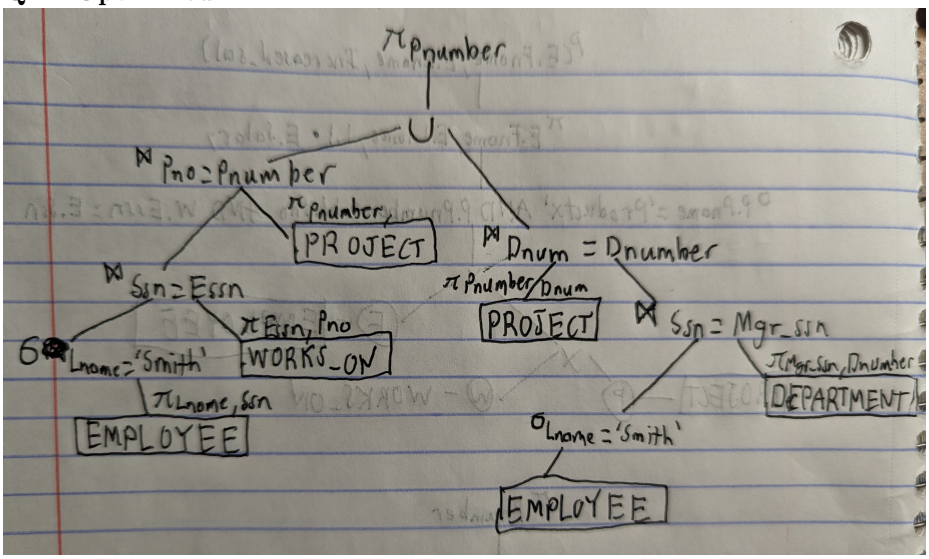


Q4A Unoptimized:



Step 1 specified a break up of any SELECT operations, and that was used to break up the two third to final SELECT operations into multiple cascading SELECT Operations. I used Step 2 and 4 to move the SELECT Operations down as far as they can, as well as combining them with their respective CARTESION PRODUCT into a JOIN operation. I followed Step 3 by moving the most restricive SELECT operations down to the bottom of the tree. Step 5 was used to move project operations as far down the tree as needed.

Q4A Optimized:



3 1.0 Point

Suppose that a disk unit has the following parameters: seek time s = 20 msec; rotational delay rd = 10 msec; block transfer time btt = 1 msec; block size B = 2400 bytes; interblock gap size G = 600 bytes. An EMPLOYEE file has the following fields: Ssn , 9 bytes; Last_name , 20 bytes; First_name , 20 bytes; Middle_init , 1 byte; Birth_date , 10 bytes; Address , 35 bytes; Phone , 12 bytes; Supervisor_ssn , 9 bytes; Department , 4 bytes; Job_code , 4 bytes; deletion marker, 1 byte. The EMPLOYEE file has r = 30,000 records, fixed-length format, and unspanned blocking. Write appropriate formulas and calculate the following values for the above EMPLOYEE file:

(a) **Calculate the record size R (including the deletion marker), the blocking factor bfr, and the number of disk blocks b.**
Record size:
Ssn + Last_name + First_name + Middle_init + Birth_date + Address + Phone + Supervisor_ssn + Department + Job_code + deletion marker

$$= 9 + 20 + 20 + 1 + 10 + 35 + 12 + 9 + 4 + 4 + 1 = 125 \text{ bytes}$$

$bfr = \lfloor B/R \rfloor = \lfloor 2400/125 \rfloor = 19$
Number of disk blocks required $= \lceil r/bfr \rceil = \lceil 30000/19 \rceil = 1579$ blocks

(b) **Calculate the wasted space in each disk block because of the unspanned organization.**
Unused space $= B - (bfr * R) = 2400 - (19 * 125) = 25$ bytes

(c) **Calculate the transfer rate tr and the bulk transfer rate btr for this disk unit (see Appendix B in the textbook for definitions of tr and btr).**
$tr = B/btt = 2400/1 = 2400$ bytes/msec
$btr = (B/(B + G)) * tr = (2400/(2400 + 600)) * 2400 = 1920$ bytes/msec

(d) **Calculate the average number of block accesses needed to search for an arbitrary record in the file, using linear search.**
average block accesses for linear search $= b/2 = 1579/2 = 789.5$ blocks

(e) **Calculate in msec the average time needed to search for an arbitrary record in the file, using linear search, if the file blocks are stored on consecutive disk blocks and double buffering is used.**
Average time to search for an arbitrary record = average blocks * average time $= b/2 * (s + rd + (k * B/btr)) = 1579/2 * (20 + 10 + (1579 * (2400/1920))) = 1581960.625$ msec