Exercises

(Course: Database Management Systems)

Chapter 3

Algorithms for Query Processing and Optimization

1. Exercise 19.13 in the text book ("Fundamentals of Database Systems- 6th Edition", Elmasri et al.)

Consider SQL queries Ql, Q8, QIB, Q4 in Chapter 4, and Q27 in Chapter 5.

Q1: **SELECT** FNAME, LNAME, ADDRESS
FROM EMPLOYEE, DEPARTMENT
WHERE DNAME='Research' **AND** DNUMBER=DNO;

Q8: **SELECT** E.FNAME, E.LNAME, S.FNAME, S.LNAME
FROM EMPLOYEE **AS** E, EMPLOYEE **AS** S
WHERE E.SUPERSSN=S.SSN;

QIB: **SELECT** E.FNAME, E.NAME, E.ADDRESS
FROM EMPLOYEE E, DEPARTMENT D
WHERE D.NAME='Research' **AND** D.DNUMBER=E.DNUMBER;

Q4: (**SELECT DISTINCT** PNUMBER
FROM PROJECT, DEPARTMENT, EMPLOYEE
WHERE DNUM=DNUMBER **AND** MGRSSN=SSN **AND** LNAME='Smith')
UNION
(**SELECT** DISTINCT PNUMBER
FROM PROJECT, WORKS_ON, EMPLOYEE
WHERE PNUMBER=PNO **AND** ESSN=SSN **AND** LNAME='Smith');

Q27: **SELECT** PNUMBER, PNAME, **COUNT** (*)
FROM PROJECT, WORKS_ON, EMPLOYEE
WHERE PNUMBER=PNO **AND** SSN=ESSN **AND** DNO=5
GROUP BY PNUMBER, PNAME;

a. Draw at least two query trees that can represent *each* of these queries. Under what circumstances would you use each of your query trees?

b. 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 19.7.

c. For each query, compare your own query trees of part (a) and the initial and final query trees of part (b).

2. Exercise 19.14 in the text book ("Fundamentals of Database Systems- 6th Edition", Elmasri et al.)

A file of 4096 blocks is to be sorted with an available buffer space of 64 blocks. How many passes will be needed in the merge phase of the external sort-merge algorithm?

3. Given the following relations:

EMPLOYEE(ename, ssn, bdate, address, sex, salary, dno)

PROJECT(pname, pnumber, plocation)

WORKS_ON(essn, pno, hours)

and the query:

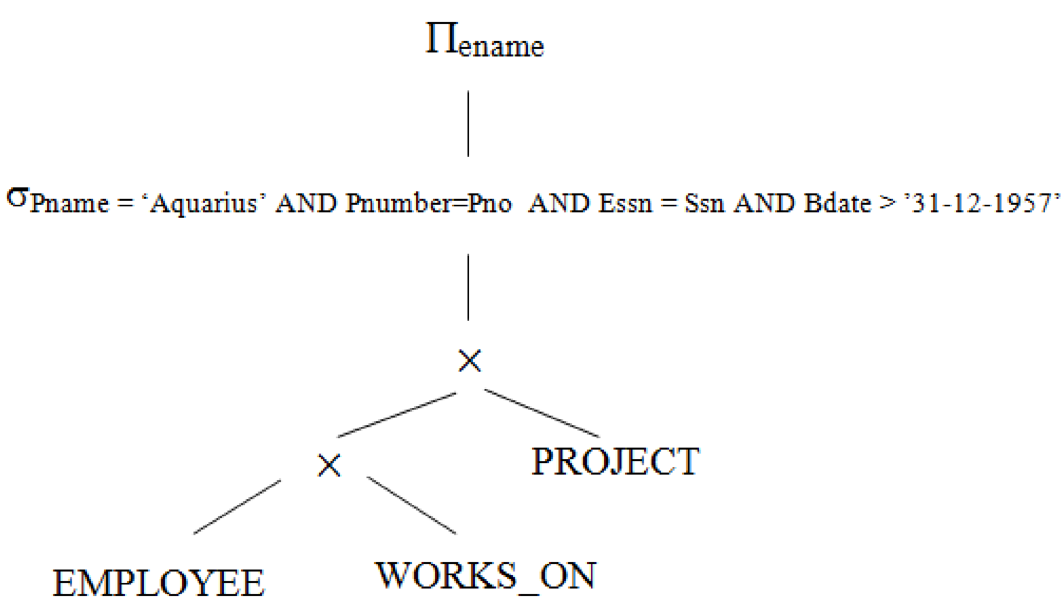
"Find the names of the employees whose birthdates are after 1957 and currently work for the project ***Aquarius****"*

Apply the heuristic optimization transformation rules to find an efficient query execution plan for the above query, which is described by the following query tree.

SELECT ename

FROM EMPLOYEE, PROJECT, WORKS_ON

WHERE pname = 'Aquarius' **AND** pnumber = pno **AND** essn = ssn **AND** bdate > '31-12-1957';



4. Given the three following relations:

SUPPLIER (Supp#, Name, City, Specialty)

PROJECT (Proj#, Name, City, Budget)

ORDER (Supp#, Proj#, Part-name, Quantity, Cost)

and the SQL query:

SELECT SUPPLIER.Name, PROJECT.Name

FROM SUPPLIER, ORDER, PROJECT

WHERE SUPPLIER.City = 'New York City' **AND** PROJECT.Budget > 10000000 **AND**

SUPPLIER.Supp# = ORDER.Supp# **AND** ORDER.Proj# = Project.Proj#

a. Write the relational algebraic expression that is equivalent to the above query and draw a query tree for the expression.
b. Apply the heuristic optimization transformation rules to find an efficient query execution plan for the above query. Assume that the number of the suppliers in New York is larger than the number of the projects with the budgets more than 10000000$ (i.e. the size of select(suppliers in New York) > the size of select (project with budget > 10 000 000$).

5. Suppose that we have the EMPLOYEE & DEPARTMENT file described as follows:

The **EMPLOYEE** file has r_E = 10000 records stored in b_E = 2000 disk blocks with the blocking factor bfr_E = 5 records/block and the following access paths:

i/ A clustering index on SALARY, with levels x_{SALARY} = 3 and first-level (base level) index blocks $bl1_{SALARY}$ = 50 and average selection cardinality s_{SALARY} = 20.

ii/ A secondary index on the key attribute SSN, with x_{SSN} = 4 (s_{SSN} = 1) and first-level (base level) index blocks $bl1_{SSN}$ = 50.

iii/ A secondary index on the nonkey attribute DNO, with x_{DNO} = 2 and first-level (base level) index blocks $bl1_{DNO}$ = 4. There are d_{DNO} = 125 distinct values for DNO, so the selection cardinality of DNO is $s_{DNO} = (r_E/d_{DNO}) = 80$.

iv/ A secondary index on SEX, with x_{SEX} = 1. There are d_{SEX} = 2 values for the SEX attribute, so the average selection cardinality is $s_{SEX} = r_E*(1/d_{SEX}) = 5000$.

The **DEPARTMENT** file consists of r_D = 125 records stored in b_D = 13 disk blocks with the blocking factor bfr_D = 10 records/block and the following access paths:

i/ A primary index on DNUMBER with $x_{DNUMBER}$ = 1 level ($s_{DNUMBER}$ = 1).

ii/ A secondary index on MGRSSN with selection cardinality s_{MGRSSN} = 1 and levels x_{MGRSSN} = 2 and first-level (base level) index blocks $bl1_{MGRSSN}$ = 2.

Determine the reasonable execution plans for each query below using cost-based optimization:

E1. SELECT *

FROM EMPLOYEE, DEPARTMENT

WHERE DNAME = 'Research' and DNUMBER = DNO;

Given: the join selectivity is js_{E1} = 1/|DEPARTMENT| = 1/r_D = 1/125 because DNUMBER is a key of DEPARTMENT; the blocking factor for the resulting join file is bfr_{ED} = 4 records per block.

E2. SELECT *

FROM EMPLOYEE as E, EMPLOYEE as S

WHERE E.SUPERSSN = S.SSN;

Given: the join selectivity is js_{E2} = 1/|EMPLOYEE| = 1/r_E = 1/10000 because SSN is a key of EMPLOYEE; the blocking factor for the resulting join file is bfr_{EE} = 2 records per block; the available buffer space in main memory is n_B = 7 blocks.