
Information Systems January 11, 2019
Master of Science in Computer Engineering

Exercise 1 (9 points)
Let's consider the following relational schema:

PAINTING (<u>PId</u>, Subject, Painter, Technique, Support, Year, Price)
MUSEUM (<u>Mid</u>, Name, Director, City, Ticket)
EXIBITION (<u>EId, PId</u>,Title, MId, OpeningDate, ClosingDate)

Primary keys are underlined in the relations. Title is unique for an exhibition. Moreover, MId in EXIBITION is foreign key of MUSEUM, and PId in EXIBITION is foreign key of PAINTING. Techniques in PAINTING are Oil and Watercolour. Support in PAINTING is one of: wall, paper, canvas and wood. A painting appears in at least one exhibition. A museum appears in at least one exhibition.

Assume that:

$n_{PAINTING}$ =16.000	Val (Technique, PAINTING) = 2
n_{MUSEUM} = 125	Val (Support, PAINTING) = 4
$n_{EXIBITION}$ = 50.000	Val (City, MUSEUM) = 5
	Val (Title, EXIBITION) = 1000

Given the query:
Title of exibitions in Parise with at least one oil painting painted on wood panel.

1) express the query as a relational-algebra expression;
2) show the basic steps of the query optimization process in terms of relational-algebra expression transformations
3) give an efficient strategy for computing the query.

Point 1

Let E, M and P denote EXIBITION, MUSEUM and PAINTING, respectively.
Π_{Title} ($\sigma_{City=Parise\ and\ Technique=oil\ and\ Support=wood}$ (E |X| M |X| P)

Point 2

Push selection down
Π_{Title} (E |X| ($\sigma_{City=Parise}$(M) |X| ($\sigma_{Technique=oil}$ ($\sigma_{Support=wood}$ (P)))

Push projection down
Π_{Title} (($\Pi_{Title,PId,MId}$ E) |X| (Π_{MId} ($\sigma_{City=Parise}$(M)) |X| (Π_{PId}($\sigma_{Technique=oil}$ ($\sigma_{Support=wood}$ (P))))

Estimate of size and different values for the new relations.

Let $E' = \Pi_{PId,Mid,Title}$ (E)
$n_{E'}$ = min($n_{E,}$ Val(PId, E)*Val(PMId, E)*Val(Title,E)) = min(50.000, 16000 * 125 * 1000) = 50.000

Let $M' = \sigma_{City=Parise}$(M)
$n_{M'} = n_M$ / V(City, M) = (125/5) = 25

Let $M'' = \Pi_{MId}$ (M')
$n_{M''} = n_{M'}$ = 25 MId is a key
Val(MId, M") = 25

Table P. Ordering of select.
Let $P' = \sigma_{Support=wood}$(P)
$n_{P'} = n_P$ / Val(Support, P) = (16.000/4) = 4.000
Val(Technique, P') = 2

Let $P'' = \sigma_{Technique=oil}$(P)
$n_{P''} = n_P$ / Val(Technique, P) = (16.000/2) = 8.000
Val(Support, P'') = 4

Selection on Support executed first than selection on Technique.
Let $P''' = \sigma_{Technique=oil}$(P')
$n_{P'''} = n_{P'}$ / Val(Technique, P') = (4.000/2) = 2.000

Let $P\wedge = \Pi_{PId}$ (P''')
$n_{P\wedge} = n_{P'''}$ = 2000 PId is a key
Val(PId, P^) = 2000

Point 3

Π_{Title} (E' |*X*| M'' |*X*| P^)

We estimate the size of different combinations of join.

T1 = (M'' |X| P^) cartesian product
Number of records in the result: 25*2000 = 50.000
n_{T1} = 50.000

T2 = (E' |X| M")
Mid in E' is not foreign key of M''
n_{T2} < 50000
n_{T2} = min(($n_{E'}$ / Val(Mid, E') * $n_{M''}$) , ($n_{M''}$ / Val(Mid, M'') * $n_{E'}$))=
min(50.000/125 * 25, 4/4 *50000) = 10.000
n_{T2} = 10.000

T3 = (E' |X | P^)
PId in E' is not foreign key of P^

$n_{T3} < 50000$

n_{T3} = min(($n_{E'}$ / Val(Pid, E') * $n_{P\wedge}$) , ($n_{P\wedge}$ / Val(Pid, P^) * $n_{E'}$))=
min(50.000/16.000 * 2000, 2000/2000 *50000) = 6.250

n_{T3} = 6.250

The best ordering of join is: (E' |X| P^) |X| M")

An efficient strategy for solving the query is:

Π_{Title} (($\Pi_{Title,PId,MId}$ E |X| Π_{MId} ($\sigma_{City=Parise}$(M)) |X| Π_{PId}($\sigma_{Technique=oil}$ ($\sigma_{Support=wood}$ (P))))

Exercise 2 (6 points)

Consider the following schedule of concurrent transactions:
S: r3(z) r2(z) w1(x) r3(y) r1(z) w3(x) w2(y) w1(y) r1(x)

1) Show if S is conflict serializable (CSR) or view serializable (VSR). Explain why. If serializable, show equivalent serial schedules.

2) Apply the rigorous two-phase locking protocol to the schedule.
3) Apply the timestamp-ordering protocol to the schedule, assuming that aborted transactions are immediately restarted.

Point 1.

x: w1 w3 r1 t1 precede t3 e t3 precede t1 non e' CSR
Non e' VSR perche' non ci sono blind write

Point 2.

r3(z) t3: slock(z) ok
r2(z) t2: slock(z) ok
w1(x) t1: xlock(x) ok
r3(y) t3: slock(y) ok
r1(z) t1: slock(z) ok
w3(x) t3:xlock(x) t3 messa in attesa di t1
w2(y) t2: xlock(y) t2 messa in attesa di t3
w1(y) t1: xlock(y) t1 messa in attesa di t3 DEADLOCK
r1(x)

Point 3

t2 abortita w2(y) rieseguita come t4
t1 abortita w1(y) rieseguita come t5

RTS(x) = 0 5
WTS(x) = 0   1   3   5
RTS(y) = 0   3
WTS(y) = 0   4   5
RTS(z) = 0   3   4
5
WTS(z) = 0

Exercise 3 (6 points)
Consider an empty B+-tree with m= 5.

1) Show the B+-tree after the insertion of the following values of the search key:
3   83   9   4   30   15   33   27   24   31   7

3<=p<=5
2<=k<=4

3 4   9   83

          30
3 4 9   30 83

             30
3 4   9 15   30 33 83

                15   30
3 4   9   15 27   30 33 83

                15   30
3 4 7   9   15 24 27   30   31 33 83

2) Show the form of the B+-tree after each operation of the sequence:
Insert 2; Insert 36; Delete 4.

               7   15   30
2 3 4   7   9   15 24 27   30   31 33 83
----
               7   15   30   36
2 3 4   7   9   15 24 27   30   31 33   36 83

----
           7   15   30   36
2 3   7   9   15 24 27   30   31 33   36 83

Exercise 4 (9 points)

Let r=(A,B,C), with primary key A uniformly distributed on the interval [1; 10.000.000].

Assume
nr = 1.200.000 number of records in the relation
Lr = 100 byte size of a record (fixed length records)
LA = 8 byte size of attribute A
Lp = 4 byte size of a pointer
Lb = 1000 byte size of a block
Heap file organization on A.

1. Show (a) the minimum and (b) the maximum height of a B+tree tree index on search-key A.

2. Outline the steps in answering the following queries, the **best strategy** and the **cost** in terms of number of block transfers from disk in case (a):

1) select * from r where A=xxx;
2) select * from r where 100.000 <= A < 500.000;
3) select * from r where B=xxx

Point 1.

Blocking factor index = IntInf ((1000-4)/12) = 83
84 max pointers

IntSup(84/2) = 42 min pointers

H min = IntSup(log84(1.200.000)) = 4
H max = IntSup(log42(1.200.000)) = 4

Point 2.

2.1
select * from r where A=xxx;
Cost= 4 +1 = 5

2.2
select * from r where 100.000 <= A < 500.000;

number of blocks of the file
1000/100 = 10 blocking factor file

IntSup(1.200.000/10) = 120.000

Linear scan: Cost 120.000 worst case 60.000 average

Selectivity factor = 400.000 / 10.000.000 = 1/25

Nfoglie: IntSup(1.200.000/83) = 14.458

Cost = 3 + IntSup(1/25(14.458)) + IntSup(1/25 (1.200.000))

Cost = 3 + 579 + 48.000 = 48.582

Best strategy using the index.

2.3
select * from r where B=xxx

Linear scan: Cost 120.000