# Query Optimization

Solutions

Page size = 4000bytes, buffer size = 10pages

Student (stdNo: integer, sName: char(15); averageGrade: integer, yearofStudy: integer,
age: integer, deptNo: integer )

Tuple length = 40bytes

Ntuples(S) = 50,000

Npages(S) = 50000*40/4000 = 500

Indexes I1: clustered B+ tree index on stdNo

Department (deptNo: integer, noFaculty: integer, deptName: char(15), deptLocation:
char(10))

Tuple length: 35bytes

Ntuples(D): 50

Npages(D) = 50 * 35 / 4000 = 1

Indexes I2: clustered B+ tree index on deptNo

I3: unclustered hash index on deptName

I4: unclustered B+tree index on noFaculty

Course ( courseNo: integer, deptNo: integer, courseTitle: char(20), slotNo: integer,
room: char(15) )

Tuple length: 50bytes

Ntuples(C): 50 * 30 = 1500

Npages(C) = 1500 * 50 / 4000 = 19

Enrolled (stdNo: integer, courseNo: integer, deptNo: integer)

Tuple length: 15bytes

Ntuples(E): 50 * 1000 * 5 = 250,000

Npages(E) = 250,000 * 15 / 4000 = 938

Indexes I5: clustered hash index on stdNo

1.

(1) Produce algebra expression

$\pi_{S.yearofStudy,\ COUNT(*)}($

$GROUP\ BY_{S.yearofStudy}($

$\pi_{S.yearofStudy}($

$\sigma_{D.deptNo=S.deptNo \wedge S.averageGrade>80 \wedge D.deptName="Computing"}(D \times S))))$

(2) Use equivalences to push down $\sigma\pi$

$\pi_{S.yearofStudy,\ COUNT(*)}($

$GROUP\ BY_{S.yearofStudy}($

$\pi_{S.yearofStudy}($

$\sigma_{D.deptNo=S.deptNo}(\pi_{D.deptNo}(\sigma_{D.deptName="Computing"}D) \times \pi_{S.yearofStudy,S.deptNo}(\sigma_{S.averageGrade>80}S))))$

(3) Determine plan

1) Pass1

| Relation | Interesting Order | Plan Description | Cost | Result |
|---|---|---|---|---|
| D | D.deptNo | *Access through clustered B+ tree index I2 on D.deptNo | | |
| | | *Access through hush index I3 on deptName finding matching tuples with projection performed | 2 | 1 |
| | | *Sequential scan D finding matching tuples on deptName with projection performed | 1 | 1 |
| S | | *Sequential scan S finding matching tuples on avergeGrad with projection performed Result=500*(10/40)*((100-80) /(100-50)) =50 | 500 | 50 |

2) Pass2

| Join Method | Outer /Inner | Plan Description | Cost | Result |
|---|---|---|---|---|
| Nested loops | D/S | *For each D tuple obtained through the sequential scan matching on deptName scan S tuples obtained through the sequential scan matching on averageGrade to find tuples matching on deptNo. Cost=1+500+50+50, result=50/50=1 | 601 | 1 |
| | S/D | *For each S tuple obtained through the sequential scan matching on deptName scan D tuples obtained through the sequential scan matching on deptName to find tuples matching on deptNo. Cost=500+1+1+1, result=50/50=1 | 503 | 1 |
| | | *For each S tuple obtained through the sequential scan matching on deptName find matching tuples of D through clustered B+ tree index on D.deptNo with selection and projection on-the-fly. Cost=500+1+1, result=50/50=1 | 502 | 1 |
| Sort merge | | *D already sorted using clustered B+ tree on D.deptNo<br>*Sort S tuples resulting from sequential scan on deptNo into L1<br>*merge D and L1 with selection and projection on-the-fly.<br>Cost=500+50+(2*2*50)+(50+1)=801, result=50/50=1 | 801 | 1 |
| | | *Sort S tuples resulting from sequential scan on deptNo into L2<br>* Sort D tuples resulting from sequential scan on deptNo into L1<br>*merge L1 and L2<br>Cost=500+50+(2*2*50)+(50+1)=801, result=50/50=1 | 801 | 1 |
| Hash | | *Partition D tuples resulting from sequential scan on deptNo into L1<br>*Partition S tuples resulting from sequential scan on deptNo into L2<br>*For each L1 tuple probe corresponding partition of L2 finding match tuples<br>Cost=(500+1)+(50+1)++3*(50+1)=705, result=50/50=1 | 705 | 1 |

Because the size of join result only one page, the “Group by” can be performed in memory and projection can be performed on-the-fly. The optimal plan is sequential scan S on deptName with projection performed on-the-fly, for each tuple resulting from the previous step find matching tuples of D through clustered B+ tree index on D.deptNo with selection and projection performed on-the-fly. The total cost is 502.

2.

(1) Produce algebra expression

$ORDER\ BY_{E.courseNo}($

$\pi_{E.courseNo,\ S.stdNo,\ S.sName}($

$\sigma_{D.deptName=\text{"Economics"} \wedge S.yearofStudy=1 \wedge E.deptNo=D.deptNo \wedge S.stdNo=E.stdNo}(E \times D \times S)))$

(2) Use equivalences to push down $\sigma\pi$

Equivalence 1

$ORDER\ BY_{E.courseNo}($

$\pi_{E.courseNo,\ S.stdNo,\ S.sName}($

$\sigma_{E.deptNo=D.deptNo}((\sigma_{S.stdNo=E.stdNo}($

$E \times (\pi_{S.stdNo,\ S.sName}(\sigma_{S.yearofStudy=1} S)))) \times \pi_{D.deptNo}(\sigma_{D.deptName=\text{"Economics"}} D)))))$

Equivalence 2

$ORDER\ BY_{E.courseNo}($

$\pi_{E.courseNo,\ S.stdNo,\ S.sName}($

$\sigma_{S.stdNo=E.stdNo}(\sigma_{E.deptNo=D.deptNo}(E \times (\pi_{D.deptNo}(\sigma_{D.deptName=\text{"Economics"}} D))) \times$

$(\pi_{S.stdNo,\ S.sName}(\sigma_{S.yearofStudy=1} S)))))$

(3) Determine plan

1) Pass1

| Relation | Interesting Order | Plan Description | Cost | Result |
|---|---|---|---|---|
| D | D.deptNo | Access through clustered B+ tree index I2 on D.deptNo | | |
| | | Access through hash index I3 on deptName finding matching tuples with projection performed | 2 | 1 |
| | | Sequential scan D finding matching tuples on deptName with projection performed | 1 | 1 |
| S | S.sdtNo | Access through clustered B+ tree index I1 on S.stdNo | | |
| | | Sequential scan S finding matching tuples on yearofStudy with projection performed Result=500*(20/40)*(1/4)=63 | 500 | 63 |
| E | E.stdNo | Access through clustered hash index I5 on E.stdNo | | |
| | | Sequential scan E | 938 | 938 |

2) Pass2 for equivalent 1

| Join Method | Outer /Inner | Plan Description | Cost | Result |
|---|---|---|---|---|
| Block Nested Loops | E/S | *For each E tuple obtained through the sequential scan scan S tuples obtained through the sequential scan matching on yearofStudy to find tuples matching on stdNo.<br>Cost=(938+500)+63+(938+63*938/(10-2))=9873<br>Result=250000*(1/4)*30/4000=469 | 9873 | 469 |
| | S/E | *For each S tuple obtained through the sequential scan matching on yearofStudy scan E tuples obtained through the sequential to find tuples matching on stdNo.<br>Cost=(500+938)+(63+938*63/(10-2))=9873<br>Result=250000*(1/4)*30/4000=469 | 9005 | 469 |
| Index Nested Loops | E/S | *For each E tuple obtained through the sequential scan scan S tuples obtained through clustered B+ tree index I1 on S.stdNo to find tuples matching on stdNo. Assume 2 I/Os to access I1.<br>Cost=938+250000*3=750938<br>Result=250000*(1/4)*30/4000=469 | 750938 | 469 |
| | S/E | *For each S tuple obtained through the sequential scan scan E tuples obtained through clustered hash index I5 on E.stdNo to find tuples matching on stdNo. Assume 1.2 I/Os to access I5<br>Cost=500+63+12500*2.2=28063<br>Result=250000*(1/4)*30/4000=469 | 28063 | 469 |
| Sort-merge | | *E and S are already sorted on stdNo<br>*merge E and S<br>Cost=938+500=1438<br>Result=250000*(1/4)*30/4000=469 | 1438 | 469 |
| Hash | | *Partion E, S<br>*join E and S<br>Cost=500+63+(938*2*2+63*2+938+63)=5442<br>Result=250000*(1/4)*30/4000=469 | 5442 | 469 |

Pass2 for equivalent 2

| Join Method | Outer /Inner | Plan Description | Cost | Result |
|---|---|---|---|---|
| Block Nested Loops | E/D | *For each E tuple obtained through the sequential scan scan D tuples obtained through the sequential scan matching on deptName to find tuples matching on deptNo.<br>Cost=938+1+1+1=941<br>Result=250000*(1/50)*10/4000=13 | 941 | 13 |
| | D/E | *For each D tuple obtained through the sequential scan matching on deptName scan E tuples obtained through the sequential scan to find tuples matching on deptNo.<br>Cost=1+938=939<br>Result=250000*(1/50)*10/4000=13 | 939 | 13 |
| Index Nested Loops | E/D | *For each E tuple obtained through the sequential scan scan D tuples obtained through clustered B+ tree index I2 on D.deptNo to find tuples matching on stdNo. Assume 2 I/Os to access I2<br>Cost=938+2=940<br>Result=250000*(1/50)*10/4000=13 | 940 | 13 |
| Sort-merge | | *D is already sorted on deptNo<br>*Sort E on deptNo<br>*merge E and D<br>Cost=1+938*4*2+938=8443<br>Result=250000*(1/50)*10/4000=13 | 8443 | 13 |
| Hash | | *Partion E, D<br>*join E and D<br>Cost=1+1+(938*2*2+1*2+938+1)=5442<br>Result=250000*(1/50)*10/4000=13 | 4695 | 13 |

2) Pass3

| Join Method | Outer /Inner | Plan Description | Cost | Result |
|---|---|---|---|---|
| Block Nested Loops | (E×S)/D | *For each (E×S) tuple obtained from pass2 scan D tuples obtained through the sequential scan matching on deptName to find tuples matching on deptNo.<br>Cost=1438+3=1441<br>Result=250000*(1/4)*(1/50)*25/4000=8 | 1441 | 8 |
| | (E×D)/S | * For each (E×D) tuple obtained from pass2 scan S tuples obtained through the sequential to find tuples matching on stdNo.<br>Cost=939+500+63+63*13/(10-2)=1441<br>Result=250000*(1/4)*(1/50)*25/4000=8 | 1628 | 8 |
| Index Nested Loops | (E×S)/D | * For each (E×S) tuple obtained from pass2 scan D tuples obtained through clustered B+ tree index I2 on D.deptNo to find tuples matching on deptNo.<br>Cost=1438+2=1440<br>Result=250000*(1/4)*(1/50)*25/4000=8 | 1440 | 8 |
| | (E×D)/S | * For each (E×D) tuple obtained from pass2 scan S tuples obtained through clustered B+ tree index I1 on D.stdNo to find tuples matching on stdNo.<br>Assume 2 I/Os to access I1<br>Cost=939+(250000/50)*3=1441<br>Result=250000*(1/4)*(1/50)*25/4000=8 | 15939 | 8 |
| Sort-merge | | ------ | | |
| Hash | | ------ | | |

The result table size is 8 pages. We sort it on E.courseNo with additional cost 16. The optimal plan is as follows: first, perform sort-merge join on E and S by accessing E through clustered hash index on E.stdNo and accessing S through clustered B+ tree index on S.stdNo. The selection and projection are performed on-the-fly. Second, we perform index nested loops join on the result from first step as outer and D as inner. D tuples obtained through clustered B+ tree index on D.deptNo. Final, we perform in memory sort on E.course on the result from second step. The total cost is:

Sort-merge join: Npages(E) + Npages(S) = 938 + 500 = 1438I/Os

Index nested loops join: the result from first step is pipelined into this step. Cost is access D. Because D can fit in one page, the cost is reading index 1 page plus reading data 1 page = 2I/Os.

Sort result: 16 I/Os.

Total cost is 1438 + 2 + 16 = 1456(I/Os).

3

First we evaluate the inner query:

$\pi_{D.deptNo}(\sigma_{D.noFaculty>25}D)$

Sequential scan: the result is a table, say T. cost = 1, Npages(T) = 1, Ntuples(T) = 50*(30-25)/(30-20) =25

Then we evaluate the outer query, it performs a nested loops join

$\pi_{C.courseNo,\ C.courseTitle,\ C.slotNo,\ C.room}(\sigma_{C.deptNo=T.deptNo}(C \times T))$

So the cost is: Npages(C) + Npages(T) = 19 + 1 = 20

Total cost is 1+1+20 = 22 I/Os