THE UNIVERSITY OF MELBOURNE

INFO20003 Database Systems

Xiuge Chen

Tutorial 8

2021.04.26

Agenda

1. Notice - 2min
2. Estimate cost of single-relation plans - 20min
3. Estimate cost of multi-relation plans - 35min

Notice

1. Assignment 2 has released - LMS Assessments
2. due date: **6:00pm Friday 30 April**
3. Tips:
   - Follow the submission instruction and format
   - Try SQL practice first - LMS Practice on your own / Lab
   - Might involve some SQL functions not taught - Google
   - Complex queries - break down into sub tasks - nest
   - Always check solutions manually

Estimate cost of single-relation plans

1. Consider a relation with this schema:

Employees (*eid*: integer, *ename*: string, *sal*: integer, *title*: string, *age*: integer)

Suppose that the following indexes exist:

a. An unclustered hash index on *eid*

b. An unclustered B+ tree index on *sal*

c. An unclustered hash index on *age*

d. A clustered B+ tree index on (*age*, *sal*)

Estimate cost of single-relation plans

1. Consider a relation with this schema:

Employees relation contains 10,000 pages and each page contains 20 tuples. Suppose there are 500 index pages for B+ tree indexes and 500 index pages for hash indexes. There are 40 distinct values of ***age***, ranging from 20 to 60, in the relation. Similarly, ***sal*** ranges from 0 to 50,000 and there are up to 50,000 distinct values. ***eid*** is a candidate key; its value ranges from 1 to 200,000 and there are 200,000 distinct values.

For each of the following selection conditions, compute the **Reduction Factor (selectivity)** and the **cost of the *cheapest* access path** for retrieving all tuples from Employees that satisfy the condition:

Estimate cost of single-relation plans

1. Consider a relation with this schema:

Employee: 10,000 pages, 20 tuples / page

eid: integer, 200,000 distinct values, from 1 to 200,000

ename: string,

sal: integer, 50,000 distinct values, from 0 to 50,000

title: string,

age: integer, 40 distinct values, from 20 to 60

a. An unclustered hash index (500 pages) on *eid*

b. An unclustered B+ tree index (500 pages) on *sal*

c. An unclustered hash index (500 pages) on *age*

d. A clustered B+ tree index (500 pages) on (*age*, *sal*)

Estimate cost of single-relation plans

What is RF (Reduction Factor, selectivity)?

Reduction factor (RF) associated with each predicate reflects the impact of the predicate in reducing the result size

1. Col = value
   RF = 1/NKeys(Col)
2. Col > value
   RF = (High(Col) – value) / (High(Col) – Low(Col))
3. Col < value
   RF = (val – Low(Col)) / (High(Col) – Low(Col))
4. Col_A = Col_B (for joins)
   RF = 1/ (Max (NKeys(Col_A), NKeys(Col_B)))
5. In no information about Nkeys or interval, use a “magic number”
   RF = 1/10

Estimate cost of single-relation plans

1. Consider a relation with this schema:

Employee: 10,000 pages, 20 tuples / page

eid: integer, 200,000 distinct values, from 1 to 200,000

ename: string,

sal: integer, 50,000 distinct values, from 0 to 50,000

title: string,

age: integer, 40 distinct values, from 20 to 60

a. An unclustered hash index (500 pages) on *eid*

b. An unclustered B+ tree index (500 pages) on *sal*

c. An unclustered hash index (500 pages) on *age*

d. A clustered B+ tree index (500 pages) on (*age*, *sal*)

Estimate cost of single-relation plans

a. sal > 20,000 -> $\sigma_{sal > 20000}(R)$

RF: = (High(Col) – value) / (High(Col) – Low(Col))
= (50,000 – 20,000) / (50,000 – 0)
= 0.6

There are 2 possible access paths for this query:

1. The unclustered B+ tree index on *sal:*
   Cost: = product of RFs of matching selects × (NTuples(*R*) + NPages(*I*))
   = 0.6 × ((20 × 10,000) + 500)
   = 120,300 I/Os

2. Full table scan
   Cost: Number of pages = 10,000 I/Os

Estimate cost of single-relation plans

b. Age = 25 -> $\sigma_{age = 25}$ (R)

RF: = 1/NKeys(Col) = 1 / 40

There are 3 possible access paths for this query:

1. The clustered B+ tree index on (*age*, *sal*):

   Cost: = product of RFs of matching × (NPages(*R*) + NPages(*I*))
   = 1 / 40 * (500 + 10,000)
   = 263 I/Os

2. The unclustered hash index on *age*:

   Cost: = product of RFs of matching × hash lookup cost × NTuples(*R*)
   = 1 / 40 * 2.2 * (20 * 10,000)
   = 11,000

3. Full table scan

   Cost: Number of pages = 10,000 I/Os

Estimate cost of single-relation plans

c. Age > 30 -> $\sigma_{age > 30}$ (R)

RF: = (High(Col) – value) / (High(Col) – Low(Col))

= (60 - 30) / (60 - 20)

= 0.75

There are 2 possible access paths for this query:

1. The clustered B+ tree index on (*age*, *sal*):

   Cost: = product of RFs of matching × (NPages(*R*) + NPages(*I*))

   = 0.75 * (500 + 10,000)

   = 7875 I/Os

2. Full table scan

   Cost: Number of pages = 10,000 I/Os

Estimate cost of single-relation plans

d. eid = 1000 -> $\sigma_{eid = 1000}$ (R)

RF: = 1 / NKeys(Col) = 1 / 200,000

There are 2 possible access paths for this query:

1. The unclustered hash index on *eid:*

   Cost: = product of RFs of matching × (hash lookup cost + access table) × N
   = 1 / 200,000 * 2.2 * (20 * 10,000)
   = 2.2 I/Os

2. Full table scan

   Cost: Number of pages = 10,000 I/Os

When query a single instance (based on primary key):

Cost = hash lookup cost + 1 data page access = 1.2 + 1 = 2.2

Estimate cost of single-relation plans

e. sal > 20,000 ∧ age > 30 -> $\sigma_{sal > 20,000 \wedge age > 30}$ (R)

RF: = $RF_{age > 30} * RF_{sal > 20,000}$

= 0.75 * 0.6

There are 3 possible access paths for this query:

1. The clustered B+ tree index on (*age*, *sal*):

   Cost: = product of RFs of matching × (NPages(*R*) + NPages(*I*))

   = 0.75 * 0.6 * (500 + 10,000)

   = 4725 I/Os

2. Full table scan

   Cost: Number of pages = 10,000 I/Os

3. The unclustered B+ tree index on *sal:*

   Cost: 0.6 * ((20 × 10,000) + 500) = 120,300 I/Os

Estimate cost of single-relation plans

Any questions?

2. Estimate cost of multi-relation plans

Consider the following schema:

Emp (<u>eid</u>, sal, age, did)

Dept (<u>did</u>, <u>projid</u>, budget, status)

Proj (<u>projid</u>, code, report)

The number of tuples in **Emp** is 20,000 and each page can hold 20 records. The **Dept** relation has 5000 tuples and each page contains 40 records. There are 500 distinct ***dids*** in **Dept**. One page can fit 100 resulting tuples of **Dept JOIN Emp**. Similarly, **Proj** has 1000 tuples and each page can contain 10 tuples. Assuming that ***projid*** is the candidate key of Proj, there can be 1000 unique values for *projid*. The number of available buffer pages is 50, and Sort-Merge Join can be done in 2 passes. Let's assume that, if we **join Proj with Dept**, 50 resulting tuples will fit on a page.

2. Estimate cost of multi-relation plans

Consider the following query:

SELECT E.eid, D.did, P.projid

FROM Emp **AS** E, Dept **AS** D, Proj **AS** P **WHERE** E.did = D.did

AND D.projid = P.projid;

For this query, estimate the cost of the following plans, focusing on the join order and join types:

Estimate cost of multi-relation plans

2. Estimate cost of multi-relation plans

Emp: 20,000 tuples, 20 records / page

Dept: 5000 tuples, 40 records / page

did: 500 distinct

Proj: 1000 tuples,10 records / page

projid: 1000 distinct

Dept JOIN Emp: 100 tuples / page

Proj JOIN Dept: 50 tuples / page

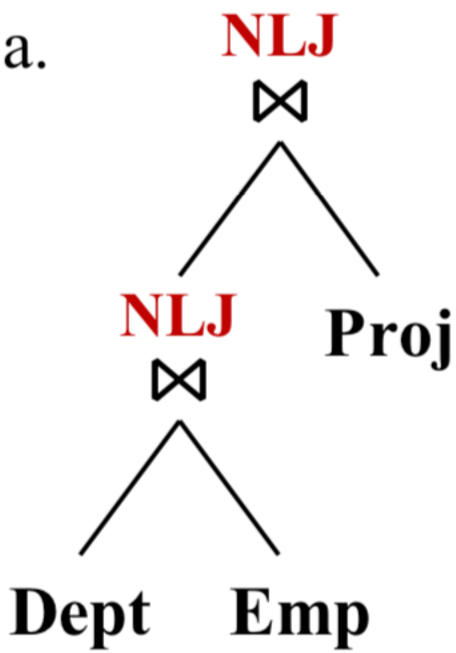
Buffer: 50 pages

Sort-Merge Join can be done in 2 passes.

Estimate cost of multi-relation plans

2. Estimate cost of multi-relation plans

a.



Total cost of NLJ

= (# of pages in outer) + (# of pages in outer × # of pages in inner)

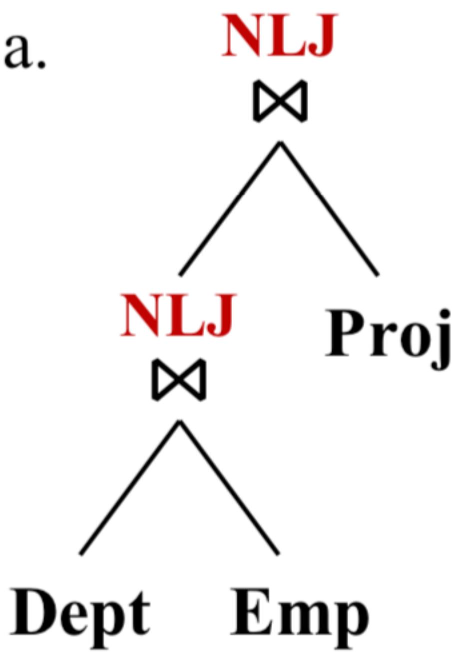
cost of scan whole outer

For each outer page

Cost of scan whole inner

2. Estimate cost of multi-relation plans

a.



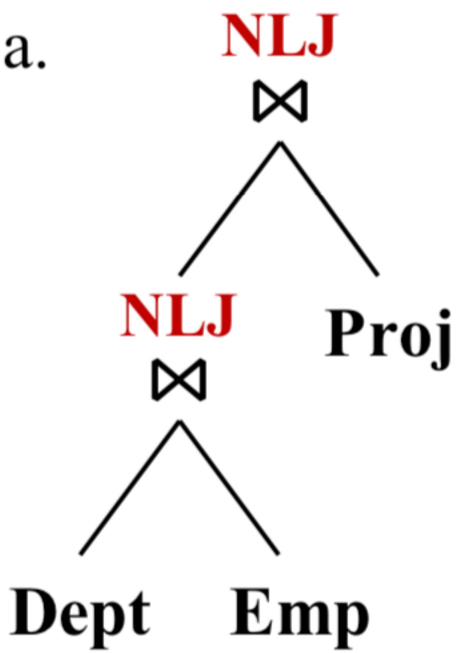
1. Cost of NLJ Dept and Emp
   = Cost of scanning Smaller + Cost of join with larger
   = Cost of scanning Dept + Cost to join with Emp
   = 125 + 125 * 1,000
   = 125,125 I/Os

2. Cost of NLJ Proj and Result of (Dept JOIN Emp)
   = Cost of join two relation (no need to scan)
   = Cost to join Result (D Join E) with Proj
   = Number of resulting pages * Npages(Proj)

Estimate cost of multi-relation plans

2. Estimate cost of multi-relation plans

a.



2. Cost of NLJ Proj and Result of (Dept JOIN Emp)

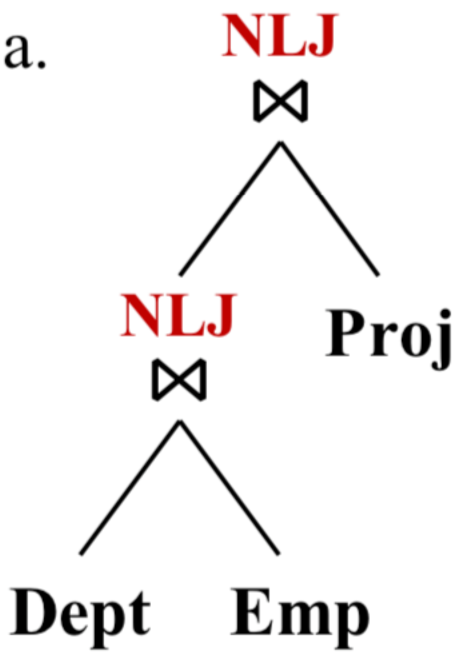
= Cost of join two relation (no need to scan)
= Cost to join Result (D Join E) with Proj
= Number of resulting pages * Npages(Proj)
= 2,000 * 100
= 200,000 I/Os

Number of resulting tuples
= 1 / NKeys(Bigger) * NTuples(Dept) * NTuples(Emp)
= 1 / 500 * 5,000 * 20,000
= 200,000 tuples

Number of resulting pages
= NTuples(r) / NTuplesAPage(r)
= 200,000 / 100 = 2000 pages

2. Estimate cost of multi-relation plans

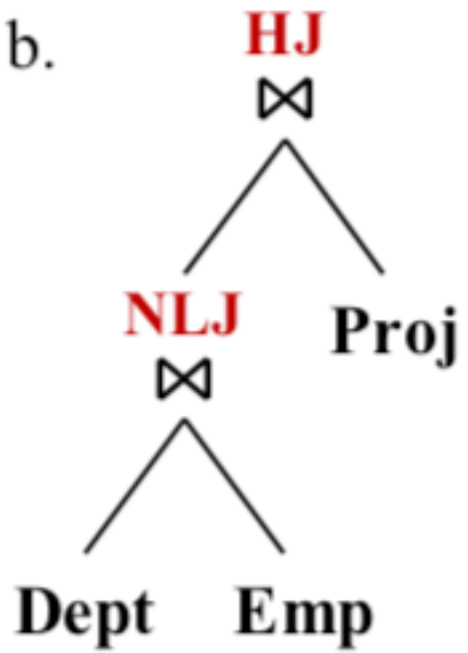
a.



Total Cost = Cost of first NLJ + Cost of second NLJ

= 125,125 + 200,000
= 325,125 I/Os

2. Estimate cost of multi-relation plans



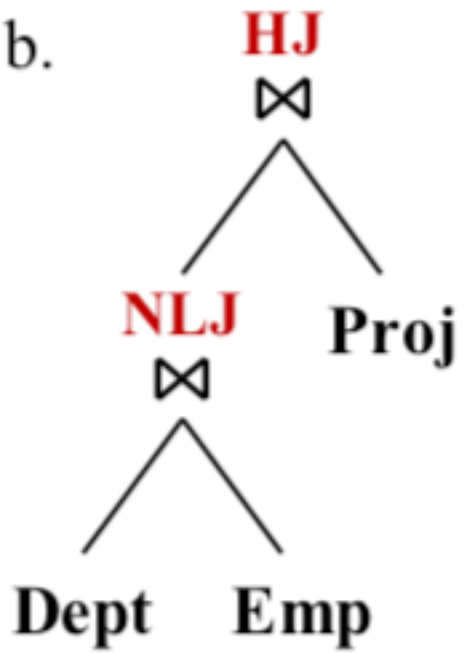
Total cost of HJ

$= 3(NPages(I) + NPages(O))$

1: Cost of scan whole table
2: Write hash results into another table
3: Scan new hash table for comparing

2. Estimate cost of multi-relation plans

b.

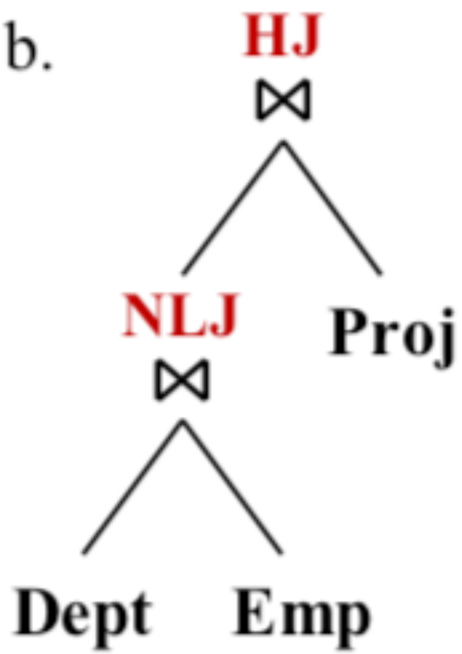


1. Cost of NLJ Dept and Emp
   = Cost of scanning Smaller+ Cost of join with larger
   = Cost of scanning Dept + Cost to join with Emp
   = 125 + 125 * 1,000
   = 125,125 I/Os

2. Cost of HJ Proj and Result of (Dept JOIN Emp)
   = 2 × NPages(Dept JOIN Emp) + 3 × NPages(Proj)
   = 2 * Number of resulting pages + 3 * Npages(Proj)

2. Estimate cost of multi-relation plans

b.



2. Cost of HJ Proj and Result of (Dept JOIN Emp)

= 2 × NPages(Dept JOIN Emp) + 3 × NPages(Proj)

= 2 * Number of resulting pages + 3 * Npages(Proj)

= 2 * 2,000 + 3 * 100

= 4,300 I/Os

Number of resulting tuples

= 1 / NKeys(Bigger) * NTuples(Dept) * NTuples(Emp)

= 1 / 500 * 5,000 * 20,000

= 200,000 tuples

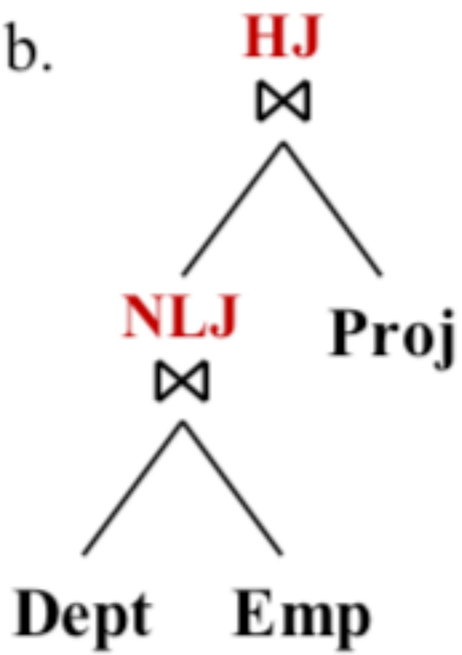
Number of resulting pages

= NTuples(r) / NTuplesAPage(r)

= 200,000 / 100 = 2000 pages

2. Estimate cost of multi-relation plans

b.

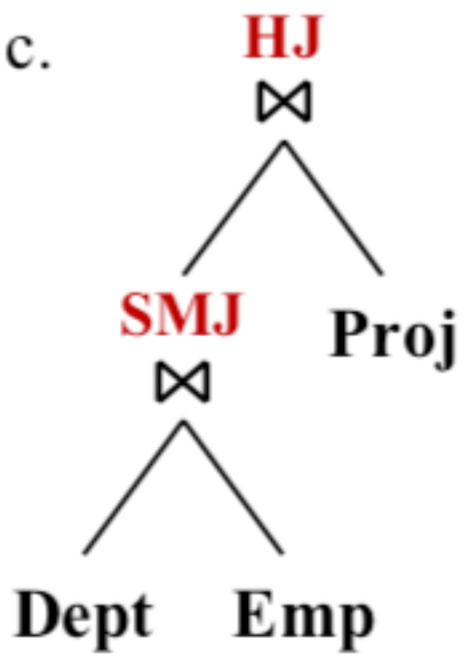


Total Cost = Cost of first NLJ + Cost of second HJ

= 125,125 + 4,300
= 129,425 I/Os

Estimate cost of multi-relation plans

2. Estimate cost of multi-relation plans



Total cost of SMJ

= Cost of sorting R + Cost of sorting S + Cost of JOIN sorted R and S

Cost of sorting R

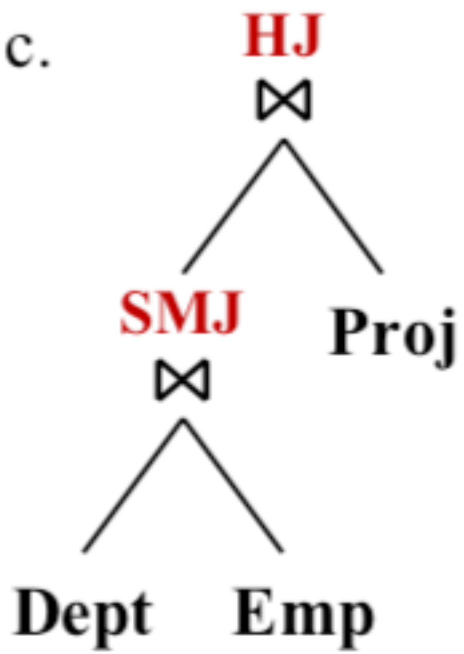
= 2 × # of passes × # of pages of R

1: Cost of scan whole table
2: Write sorted results into another table

Cost of JOIN sorted R and S = NPages(R) + NPages(S)

Scan new sorted table

2. Estimate cost of multi-relation plans



1. Cost of SMJ Dept and Emp

   = Cost of sorting Dept + Cost of sorting Emp + Cost of join sorted Dept and Emp

   = 2 × NPasses × NPages(Dept) + 2 × NPasses × NPages(Emp) + NPages(Dept) + NPages(Emp)

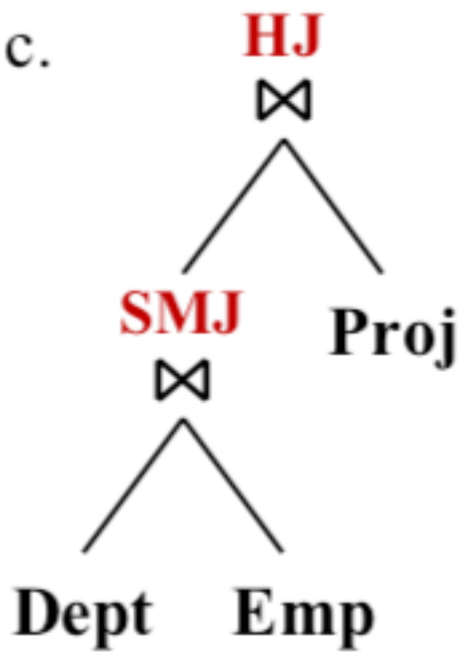
   = 2 * 2 * 125 + 2 * 2 * 1,000 + 125 + 1,000
   = 500 + 4,000 + 1,125
   = 5,625 I/Os

2. Cost of HJ Proj and Result of (Dept JOIN Emp)

   = 2 × NPages(Dept JOIN Emp) + 3 × NPages(Proj)

   = 2 * Number of resulting pages + 3 * Npages(Proj)

2. Estimate cost of multi-relation plans



2. Cost of HJ Proj and Result of (Dept JOIN Emp)
   = 2 × NPages(Dept JOIN Emp) + 3 × NPages(Proj)
   = 2 * Number of resulting pages + 3 * Npages(Proj)
   = 2 * 2,000 + 3 * 100
   = 4,300 I/Os

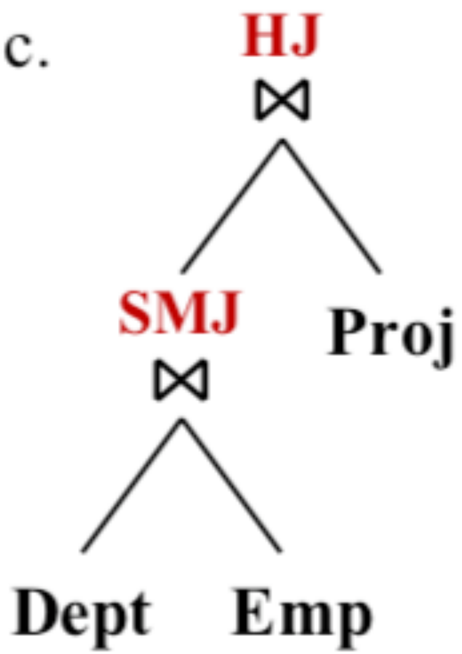
Number of resulting tuples
= 1 / NKeys(Bigger) * NTuples(Dept) * NTuples(Emp)
= 1 / 500 * 5,000 * 20,000
= 200,000 tuples

Number of resulting pages
= NTuples(r) / NTuplesAPage(r)
= 200,000 / 100 = 2000 pages

Estimate cost of multi-relation plans

2. Estimate cost of multi-relation plans

c.

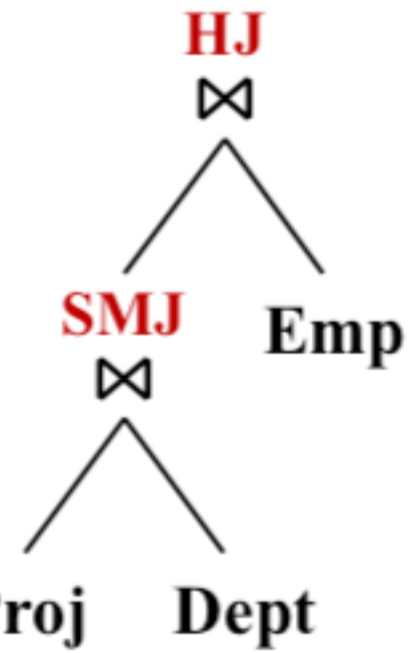


Total Cost = Cost of first SMJ + Cost of second HJ

= 5,625 + 4,300
= 9,925 I/Os

2. Estimate cost of multi-relation plans

d.



1. Cost of SMJ Dept and Proj

   = Cost of sorting Dept + Cost of sorting Proj + Cost of join sorted Dept and Proj

   = 2 × NPasses × NPages(Dept) + 2 × NPasses × NPages(Proj) + NPages(Dept) + NPages(Proj)

   = 2 * 2 * 125 + 2 * 2 * 100 + 125 + 100
   = 500 + 400 + 225
   = 1,125 I/Os

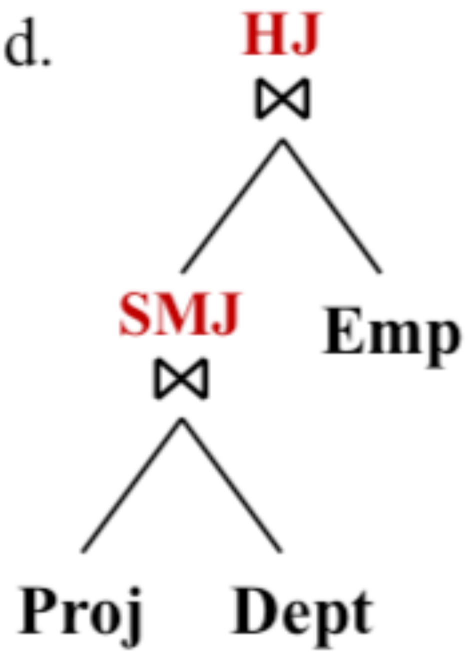
2. Cost of HJ Emp and Result of (Dept JOIN Proj)

   = 2 × NPages(Dept JOIN Proj) + 3 × NPages(Emp)

   = 2 * Number of resulting pages + 3 * Npages(Emp)

2. Estimate cost of multi-relation plans

d.



2. Cost of HJ Emp and Result of (Dept JOIN Proj)

= 2 × NPages(Dept JOIN Proj) + 3 × NPages(Emp)

= 2 * Number of resulting pages + 3 * Npages(Emp)

= 2 * 100 + 3 * 1,000

= 3,200 I/Os

Number of resulting tuples

= 1 / NKeys(Bigger) * NTuples(Dept) * NTuples(Proj)

= 1 / 1,000 * 5,000 * 1,000

= 5,000 tuples

Number of resulting pages

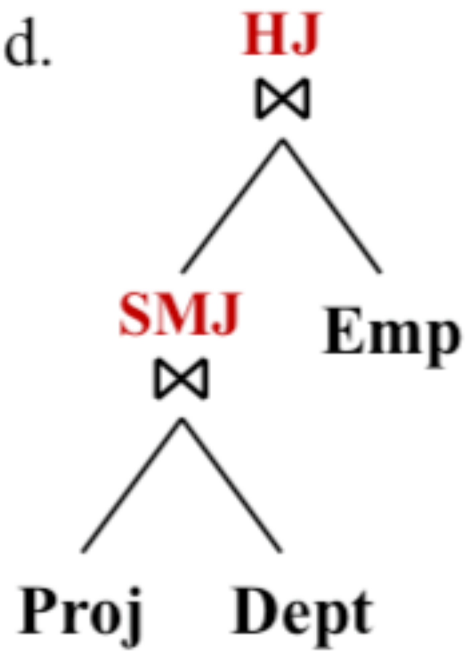
= NTuples(r) / NTuplesAPage(r)

= 5,000 / 50 = 100 pages

Estimate cost of multi-relation plans

2. Estimate cost of multi-relation plans

d.



Total Cost = Cost of first SMJ + Cost of second HJ

= 1,125 + 3,200
= 4,325 I/Os

Estimate cost of multi-relation plans

Any questions?