Big Data Engineering: Assignment 6

Leo Forster, Maximilian Prinz, Bastian Simon

1

Plan Enumeration

Subproblem 1

a) Determine the optimal join order using only "left-deep" plans. Complete and extend the following table with subplans of size two, three and four:

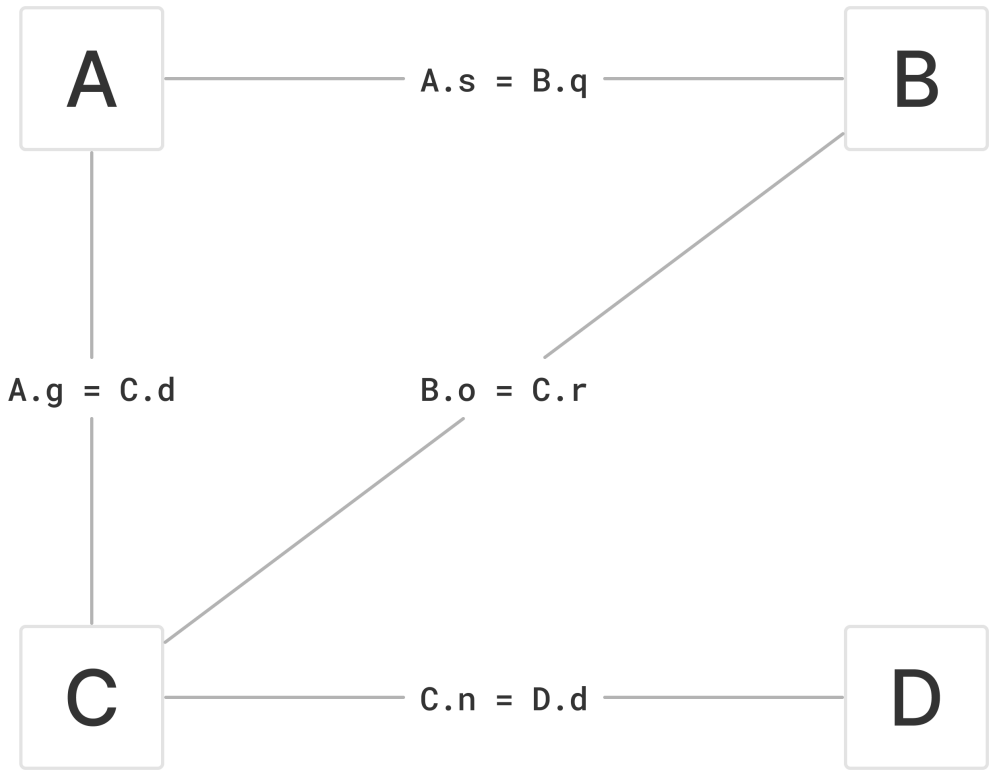
subplan	costs	result size
A	$cost(A) = 0$	50
B	$cost(B) = 0$	95
C	$cost(C) = 0$	20
D	$cost(D) = 0$	65

subplan	costs	result size
$A \bowtie B$	$cost(A) + cost(B) + cost(A \bowtie B) \implies 0 + 0 + (50 + 95) = 145$	40
$A \bowtie C$	$cost(A) + cost(C) + cost(A \bowtie C) \implies 0 + 0 + (50 + 20) = 70$	70
$A \times D$	$cost(A) + cost(D) + cost(A \times D) \implies 0 + 0 + (50 * 65) = 3250$	3250
$B \bowtie C$	$cost(B) + cost(C) + cost(B \bowtie C) \implies 0 + 0 + (95 + 20) = 115$	100
$B \times D$	$cost(B) + cost(D) + cost(B \times D) \implies 0 + 0 + (95 * 65) = 6175$	6175
$C \bowtie D$	$cost(C) + cost(D) + cost(C \bowtie D) \implies 0 + 0 + (20 + 65) = 85$	180

subplan	costs	result size
$(A \bowtie B) \bowtie C$	$cost(A \bowtie B) + cost(C) + cost((A \bowtie B) \bowtie C) \implies 145 + 0 + (40 + 20) = 205$	40
$(A \bowtie C) \bowtie B$	$cost(A \bowtie C) + cost(B) + cost((A \bowtie C) \bowtie B) \implies 70 + 0 + (70 + 95) = 235$	40
$(B \bowtie C) \bowtie A$	$cost(B \bowtie C) + cost(A) + cost((B \bowtie C) \bowtie A) \implies 115 + 0 + (100 + 50) = 265$	40
$(A \bowtie B) \times D$	$cost(A \bowtie B) + cost(D) + cost((A \bowtie B) \times D) \implies 145 + 0 + (40 * 65) = 2745$	2600
$(A \times D) \bowtie B$	$cost(A \times D) + cost(B) + cost((A \times D) \bowtie B) \implies 3250 + 0 + (3250 + 95) = 6595$	2600
$(B \times D) \bowtie A$	$cost(B \times D) + cost(A) + cost((B \times D) \bowtie A) \implies 6175 + 0 + (6175 + 50) = 12400$	2600
$(A \bowtie C) \bowtie D$	$cost(A \bowtie C) + cost(D) + cost((A \bowtie C) \bowtie D) \implies 70 + 0 + (70 + 65) = 205$	300
$(A \times D) \bowtie C$	$cost(A \times D) + cost(C) + cost((A \times D) \bowtie C) \implies 3250 + 0 + (3250 + 20) = 6520$	300
$(C \bowtie D) \bowtie A$	$cost(C \bowtie D) + cost(A) + cost((C \bowtie D) \bowtie A) \implies 85 + 0 + (180 + 50) = 315$	300
$(B \bowtie C) \bowtie D$	$cost(B \bowtie C) + cost(D) + cost((B \bowtie C) \bowtie D) \implies 115 + 0 + (100 + 65) = 280$	543
$(B \times D) \bowtie C$	$cost(B \times D) + cost(C) + cost((B \times D) \bowtie C) \implies 6175 + 0 + (6175 + 20) = 12370$	543
$(C \bowtie D) \bowtie B$	$cost(C \bowtie D) + cost(B) + cost((C \bowtie D) \bowtie B) \implies 85 + 0 + (180 + 95) = 360$	543

subplan	costs	result size
$(A \bowtie B \bowtie C) \bowtie D$	$cost(A \bowtie B \bowtie C) + cost(D) + cost((A \bowtie B \bowtie C) \bowtie D) \implies 205 + 0 + (40 + 65) = 310$	8
$(A \bowtie B \times D) \bowtie C$	$cost(A \bowtie B \times D) + cost(C) + cost((A \bowtie B \times D) \bowtie C) \implies 2745 + 0 + (2600 + 20) = 5365$	8
$(A \bowtie C \bowtie D) \bowtie B$	$cost(A \bowtie C \bowtie D) + cost(B) + cost((A \bowtie C \bowtie D) \bowtie B) \implies 205 + 0 + (300 + 95) = 600$	8
$(B \bowtie C \bowtie D) \bowtie A$	$cost(B \bowtie C \bowtie D) + cost(A) + cost((B \bowtie C \bowtie D) \bowtie A) \implies 280 + 0 + (543 + 50) = 873$	8

Here we can see that $(A \bowtie B \bowtie C) \bowtie D$ has the lowest cost and therefore is the optimal join order.



Subproblem 2

b) Draw the join graph for the given query. Label the edges with the respective join predicates.

Graph is above the Question somehow :D

Subproblem 3

c) How could you have used the join graph to exclude certain entries in the table from Task (a)?

We can omit the Cartesian products $A \times D$ and $B \times D$ because in the join graph we see that there is no direct connection between A-D and B-D.

Leo Forster
Maximilian Prinz
Bastian Simon

1. Cost-based Optimisation

1.1 (a)

1. Dataset:

Costs $B, B.s = 12$:

- scan: $300,000$
- index: $252,018.19$

=> use index

Costs $C.z \neq 17$:

- scan: $50,000$
- index: $21,015.61$

=> use index

Costs $C.c < 123$:

- scan: $50,000$
- index: $105.015,61$

=> use scan

2. Dataset:

Costs $B, B.s = 12$:

- scan: $70,000$
- index: $117,616.10$

=> use scan

Costs $C.z \neq 17$:

- scan: $150,000$
- index: $94,517.19$

=> use index

Costs $C.c < 123$:

- scan: $150,000$
- index: $472.517,19$

=> use scan

Thus, for dataset 1:

- $\sigma_{B.s=12}$ should be replaced with $\sigma\_IndexBased_{B.s=12}$
- $\sigma_{C.c<123}$ should be replaced with $\sigma\_ScanBased_{C.c<123}$
- $\sigma_{C.z\neq17}$ should be replaced with $\sigma\_IndexBased_{C.z\neq17}$

For dataset 2:

- $\sigma_{B.s=12}$ should be replaced with $\sigma\_ScanBased_{B.s=12}$
- $\sigma_{C.c<123}$ should be replaced with $\sigma\_ScanBased_{C.c<123}$
- $\sigma_{C.z\neq17}$ should be replaced with $\sigma\_IndexBased_{C.z\neq17}$

1.2 (b)

Given the rule, to use the index if: $scan(T) > index(T, p)$:

$$|T| > \log_2(|T|) + 42 \cdot \text{sel}(p) \cdot |T|$$

Solving this for $\text{sel}(p)$ we get:

$$\text{sel}(p) < \frac{|T| - \log_2(|T|)}{|T| \cdot 42}$$

With large table sizes this approximates:

$$\text{sel}(p) \approx \frac{1}{42} \approx 0.024$$

Exercise 3

1. HashMap hm1; HashMap hm2

A1: Populate hm1 with tuples from ScanBasedU

2. For all u in ScanBasedU:
3. hm1.insert(u.v,u);

A2: Filter and aggregate tuples from ScanBasedV

4. For all v in ScanBasedV:
5. if v.s >= 10 && v.s < 75:
6. avgD = calculateAverage(hm1.values(), u.d);
7. count = getCount(hm1.values());
8. aggregateTuple = Tuple(v.v, count, avgD);
9. hm2.insert(aggregateTuple.v, aggregatedTuple);

A3: Join operation using hm2

10. For all v in ScanBasedV:
11. if v.s >= 10 && v.s < 75:
12. matchingTuples = hm2.probe(v.v);
13. For all mt in matchingTuples:
14. Yield Tuple(mt.v, mt.count, mt.avgD);