Query Optimization Exercise 2

Alexandra Nordin (03650463)
Roc Reguant Comellas (03650516)

May 2014

1 Exercise 1

The canonical translation is:

$$\sigma_p(\ (Customers\ \times\ Orders)\ \times\ Lineitem)$$

where p is the following predicate:

$$l.l\_orderkey = o.o\_orderkey \wedge o.o\_custkey = c.c\_custkey \wedge c.c\_name = \text{"}Customer\#000014993\text{"}$$

For the logical optimization we've followed the next steps:
1) Breaking up conjunctive selection predicates
2) Pushing selections down
3) Introducing joins
4) Determining the join order
5) Introducing and push down projections

The first step can be seen in the Figure 1 and the final result after applying all those steps can be seen in Figure 2.

2 Exercise 2

2.1

The selectivity of $\sigma_{R1.x=c}$ can be estimated as following:

$$selectivity(\sigma_{R1.x=c}) \approx \begin{cases} 1 & \text{if R1.x is key} \\ \frac{|domain(R1.x)|}{|R1|} & \text{otherwise} \end{cases} \quad (1)$$

The motivation for this that if the domain for an attribute is high then there

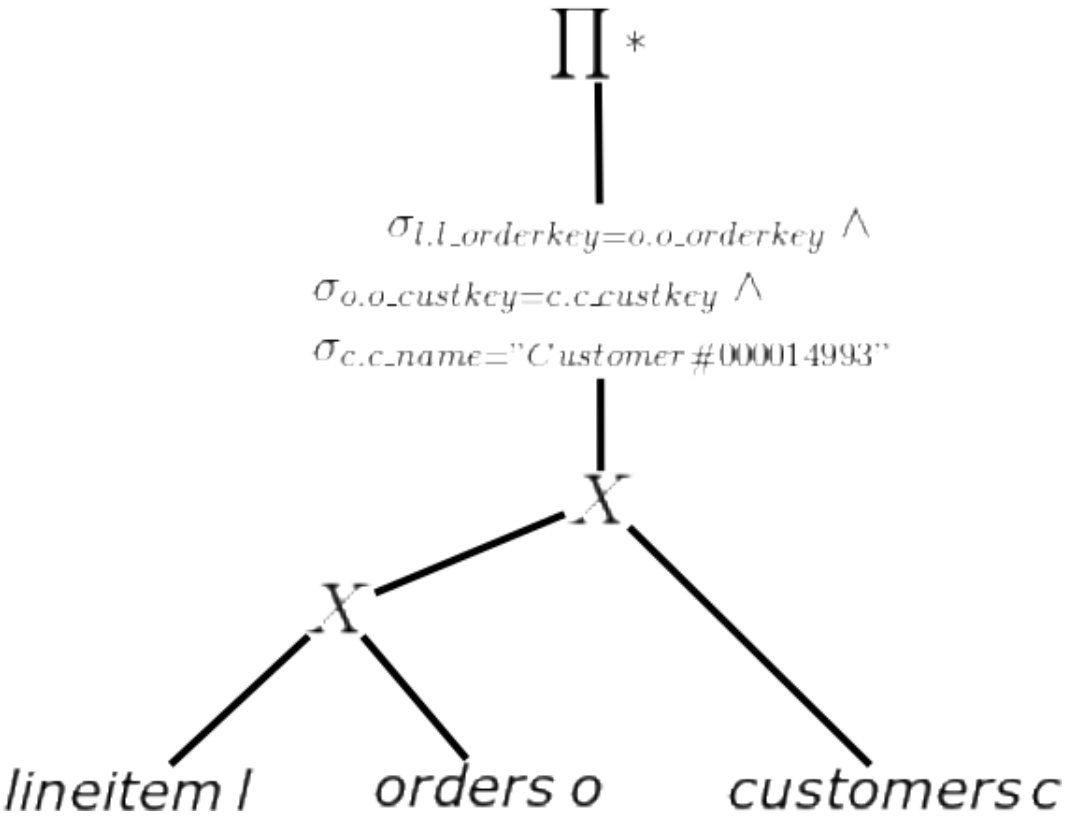

Figure 1: First step of the logical optimization

is a higher chance for a lower selectivity. In the case where R1.x is a key the domain of R1.x is equal to the cardinality giving a selectivity of 1. Therefore the estimation of the selectivity is the relation between the domain and the cardinality. The result will lie between 0 and 1.

2.2

The selectivity of $\bowtie_{R1.x=R2.y}$ can be estimated as following:

$$selectivity(\bowtie_{R1.x=R2.y}) \approx \begin{cases} 1 & \text{if R1.x and R2.y are keys} \\ \frac{|domain(R1.x)|}{|R1|} & \text{if R2.y is key} \\ \frac{|domain(R2.y)|}{|R2|} & \text{if R1.x is key} \end{cases} \quad (2)$$

If both R1.x and R2.y are keys then the selectivity will be 1. If however R2.y is a key this means that R2 will join on only one tuple, therefore the estimation of the selectivity will be the domain of R1 over the cardinality.

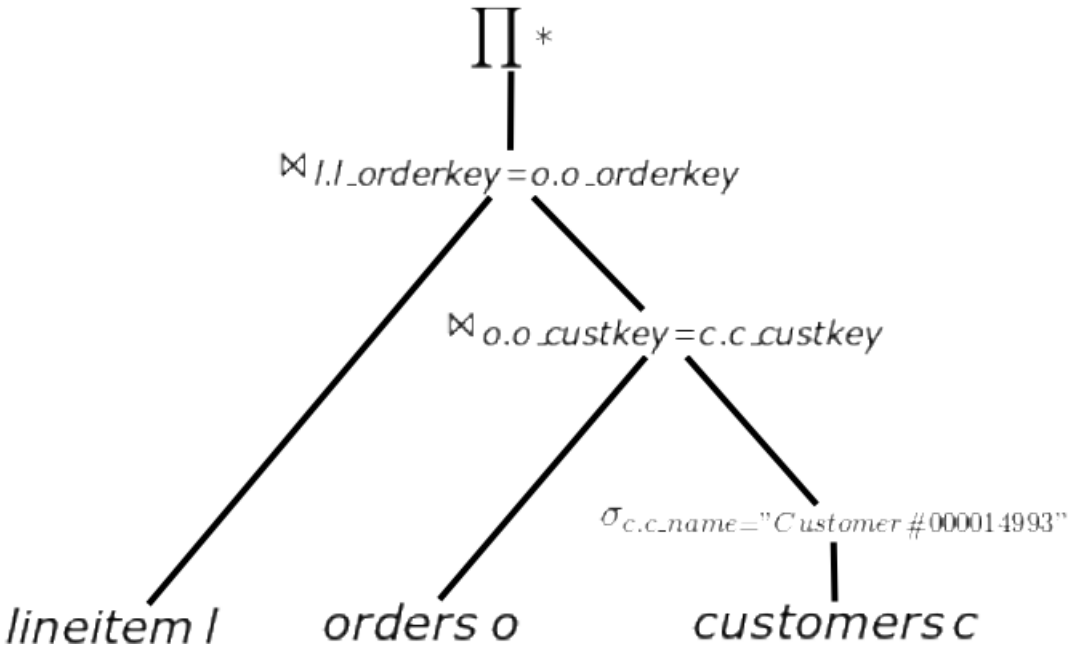

Figure 2: The last step of the logical optimization

3 Exercise 3

We assume that:
* The main memory has at least 1,000 pages space.
* All the data is aligned in the disk without spaces in the middle.
* The CPU is faster processing than the disk transferring the information.

For the Nested Loops Join:
To fetch R it takes:

$$\frac{1,000 pages}{10,000 pages/sec} + 0.01 second = 0.11 seconds$$

To fetch S it takes the number of fetches (iterations of the outer loop) plus the time of one fetch:

$$Number\ of\ fetches = 1,000 pages * 50 tuples/page = 50.000 iterations$$

$$One\ iteration\ cost: \frac{100,000 pages}{10,000 pages/second} + 0.01 seconds = 10.01 seconds$$

This makes a total of:

$$10.01 * 50000 = 500,500 seconds$$

Block Nested Loops Join:
To fetch R it takes:

$$\frac{1,000 pages}{10,000 pages/sec} + 0.01 second = 0.11 seconds$$

To fetch S it takes the number of fetches (iterations) plus the time of one fetch:

$$Number\ of\ fetches = 100,000 pages/100 page/block = 1,000 iterations$$

$$One\ iteration\ cost: \frac{100,000 pages}{10,000 pages/second} + 0.01 seconds = 10.01 seconds$$

This makes a total of:

$$10.01 * 1,000 = 10,010 seconds$$