How to understand the Join Selectivity Factor

The selectivity factor of the join tells us the proportion of tuples from the cartesian product of the two inputs that will be part of the output: i.e., |O| = SF*|R|*|S|, where O is the output, R and S the two inputs.

Most of the times, a join is executed over a PK-FK relationship. Let us consider from now on R is the table with the PK and S with the FK.

Since a FK value joins with one and only one value of the PK, the SF = 1/ |R|. Then, |O| = (1/ |R|)*|R|*|S|, which is equivalent to: |O| = |S|.

Note, however, this formula needs to be tuned up when intermediary results come into play. Assume now the input on the PK side is the result of a previous selection s'. Let's call now this input R' and to the original table R. If we apply the above formula directly, it would be wrong:

SF = 1/ |R'| and then |O| = (1/ |R'|)*|R'|*|S|, which is equivalent to |O| = |S|. If we do so we assume that the selection did not remove any PK value and all of them are available after the selection (highly unlikely). If some PK values are removed by the selection, it means that some rows from S will not join. We need to estimate how many of them. A way to do so is to weight the SF with the proportion of PK values that passed the selection:

- Consider A the attribute in R/R' participating in the join,
- ndist(A, R') as the number of distinct values of A after the selection and
- ndist(A, R) as the number of distinct vlues of A in R

Then:

SF = (1/ |R'|)*(ndist(A, R')/ndist(A,R))

Now:

|O|= (1/ |R'|)*(ndist(A, R')/ndist(A,R))*|R'|*|S| = (ndist(A, R')/ndist(A,R))*|S|. In other words, we are considering that the number of elements in S that will join is proportional to the number of values of A left after the selection over R. Note that doing so, we assume an uniform distribution of values, which is an assumption typically made by DBs.

As general rule, when working with intermediate results, you need to take this formula as basis and fine tune it to consider how many elements will participate in the join. Thus**, in the presence of previous operations or NULLs (since NULL values do not join), you need to estimate the proportion of values reaching the join**.