

# Query processing cost formulae

## Legend

Symbol	Description
<b>NKeys(Col)</b>	The number of distinct values of column Col
<b>High(Col)</b>	The highest value of column Col
<b>Low(Col)</b>	The lowest value of column Col
<b>NTuples(R)</b>	The number of tuples of relation R
<b>NPages(R)</b>	The number of pages of relation R
<b>NPages(I)</b>	The number of pages of index I
<b>Height(I)</b>	The height of index I
$\prod RF_i$	The product of all reduction factors
$\prod NTuples(R_i)$	The product of the numbers of tuples of all relations taking part in a join
$\text{ceil}(1 + \log_{B-1} \text{ceil}(NPages(R)/B))$	The number of passes for sorting

## 1. Reduction factor (Selectivity)

a. Col = value

$$RF = 1/NKeys(Col)$$

b. Col > value

$$RF = (High(Col) - value) / (High(Col) - Low(Col))$$

c. Col < value

$$RF = (val - Low(Col)) / (High(Col) - Low(Col))$$

d. Col\_A = Col\_B (for joins)

$$RF = 1/ (\text{Max} (NKeys(Col\_A), NKeys(Col\_B)))$$

e. In no information about NKeys, use a “magic number” 1/10

$$RF = 1/10$$

## 2. Result size calculations

a. Single table

$$\text{Result\_size} = NTuples(R) * \prod RF_i$$

b. Joins

$$\text{Result\_size} = \prod NTuples(R_i) * \prod RF_i$$

### 3. Indexing

#### a. B+-tree index

- i. Just a single tuple (selection over a primary key)

$$\text{Cost} = \text{Height}(I) + 1$$

- ii. Clustered index (multiple tuples)

$$\text{Cost} = (\text{NPages}(I) + \text{NPages}(R)) * \prod \text{RF}_i$$

- iii. Unclustered (multiple tuples)

$$\text{Cost} = (\text{NPages}(I) + \text{NTuples}(R)) * \prod \text{RF}_i$$

#### b. Hash Index

- i. Just a single tuple (selection over a primary key)

$$\text{Cost} = 1.2 + 1 = 2.2$$

- ii. Clustered index (multiple tuples)

$$\text{Cost} = (\text{NPages}(R)) * \prod \text{RF}_i * 2.2$$

- iii. Unclustered index (multiple tuples)

$$\text{Cost} = (\text{NTuples}(R)) * \prod \text{RF}_i * 2.2$$

### 4. Sequential Scan (i.e. Heap Scan)

$$\text{Cost} = \text{NPages}(R)$$

## 5. Joins (between relations R and S, R = outer, S = inner)

### a. NLJ

#### i. Tuple-oriented NLJ

$$\text{Cost} = \text{NPages}(\text{R}) + \text{NTuples}(\text{R}) * \text{NPages}(\text{S})$$

#### ii. Page-oriented NLJ

$$\text{Cost} = \text{NPages}(\text{R}) + \text{NPages}(\text{R}) * \text{NPages}(\text{S})$$

#### iii. Block-oriented NJL (for block\_size B)

$$\text{Cost} = \text{NPages}(\text{R}) + \text{ceil}(\text{NPages}(\text{R}) / (B - 2)) * \text{NPages}(\text{S})$$

#### iv. Index NLJ

$$\text{Cost} = \text{NPages}(\text{R}) + \text{NTuples}(\text{R}) * \text{cost of a single tuple of S}$$

### b. Hash Join

$$\text{Cost} = \text{NPages}(\text{R}) + \text{NPages}(\text{S}) + 2 * (\text{NPages}(\text{R}) + \text{NPages}(\text{S}))$$

**Note:**  $2 * (\text{NPages}(\text{R}) + \text{NPages}(\text{S}))$  is for partitioning and is optional if one relation fits entirely in memory. In that case only hashing will happen.

### c. Sort-Merge Join (for block\_size B)

$$\text{Cost}_{\text{SMJ}} = \text{NPages}(\text{R}) + \text{NPages}(\text{S}) +$$

$$2 * \text{NPages}(\text{R}) * \text{ceil}(1 + \log_{B-1} \text{ceil}(\text{NPages}(\text{R}) / B)) +$$

$$2 * \text{NPages}(\text{S}) * \text{ceil}(1 + \log_{B-1} \text{ceil}(\text{NPages}(\text{S}) / B))$$

c.1 Improvement: do not sort completely, but merge sorted runs directly (both formulae will be accepted!)

$$\text{Cost}_{\text{SMJ\_improved}} = \text{Cost}_{\text{SMJ}} - 2 * (\text{NPages}(\text{R})) - 2 * (\text{NPages}(\text{S}))$$