Database Administration

José Orlando Pereira

Departamento de Informática
Universidade do Minho

Roadmap

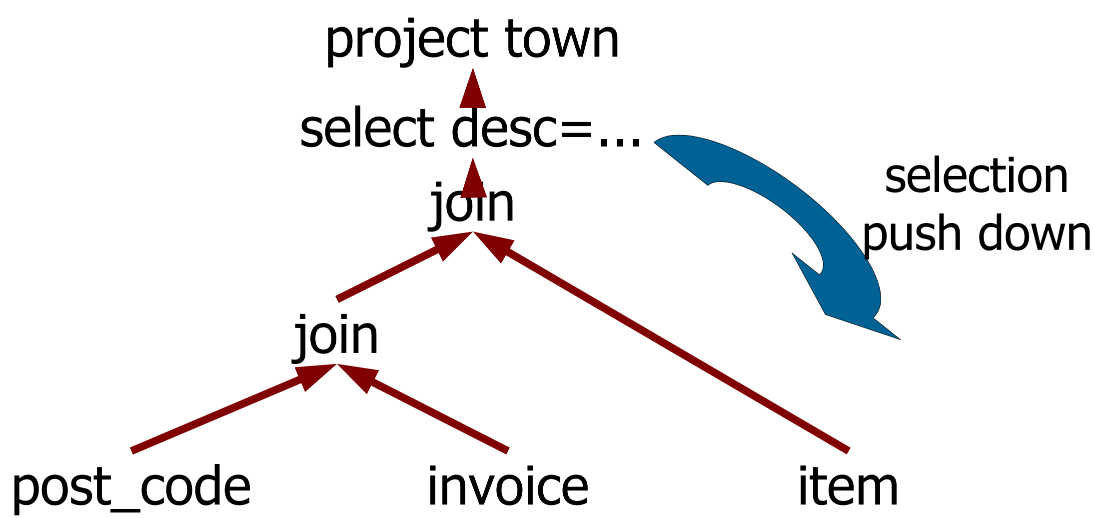
- What physical operators exist for each logical operation?

- How are physical operators implemented and composed?

- <u>How is the best plan found</u>?

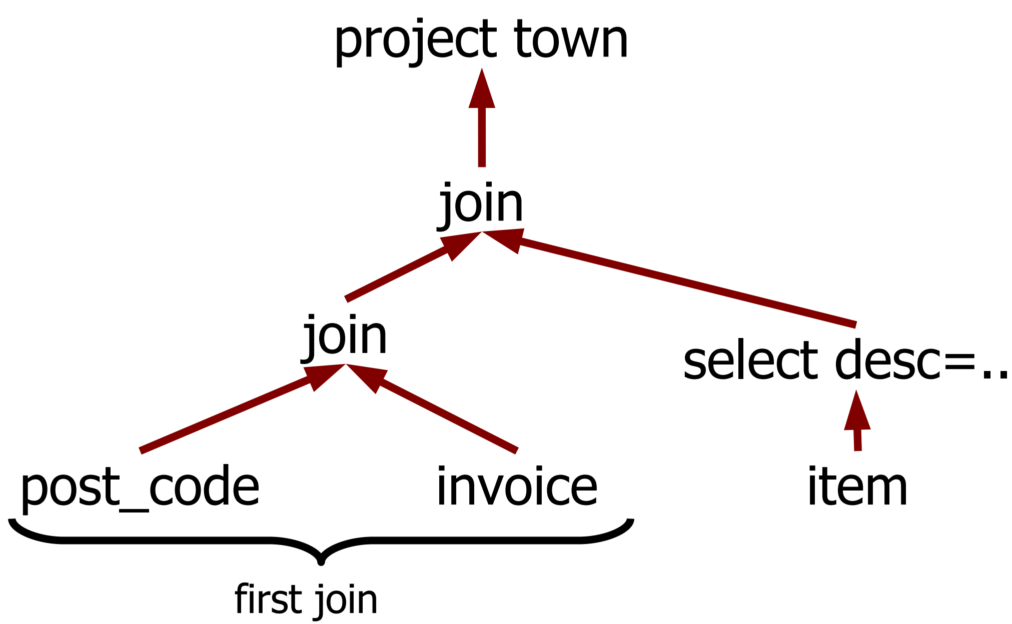
Optimization

select town from
 post_code natural join invoice
 natural join item where desc=...



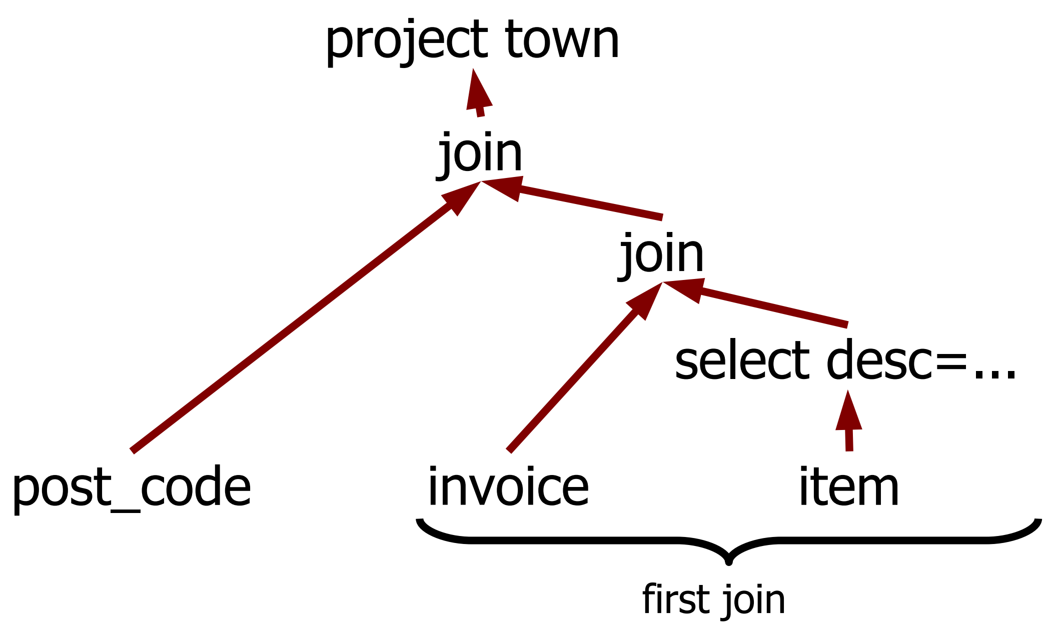
Optimization

select town from
post_code natural join invoice
natural join item where desc=...



Optimization

select town from
 post_code natural join invoice
 natural join item where desc=...



Roadmap

- <u>How to estimate the cost of a plan?</u>

- How to find alternative plans?

Cost estimation

- Tradeoff beween:
  - Actually executing the query and measuring what resources it consumes and how long it takes to execute

    vs
  - **Estimate that can be computed quickly**

    vs
  - Considering that all queries have the same cost
- We don’t want to know what is the actual cost
- We want to know which alternative costs less
  - Use a <u>model</u> that monotonously approximates real cost

Example: Simple sequential scan

- Assumption of main cost factor:
  - Number of disk block I/O operations
- <u>Cost model</u> as a simple equation:
  - $C = C_0 + C_1 N_{blocks}$

Known from the size of the file

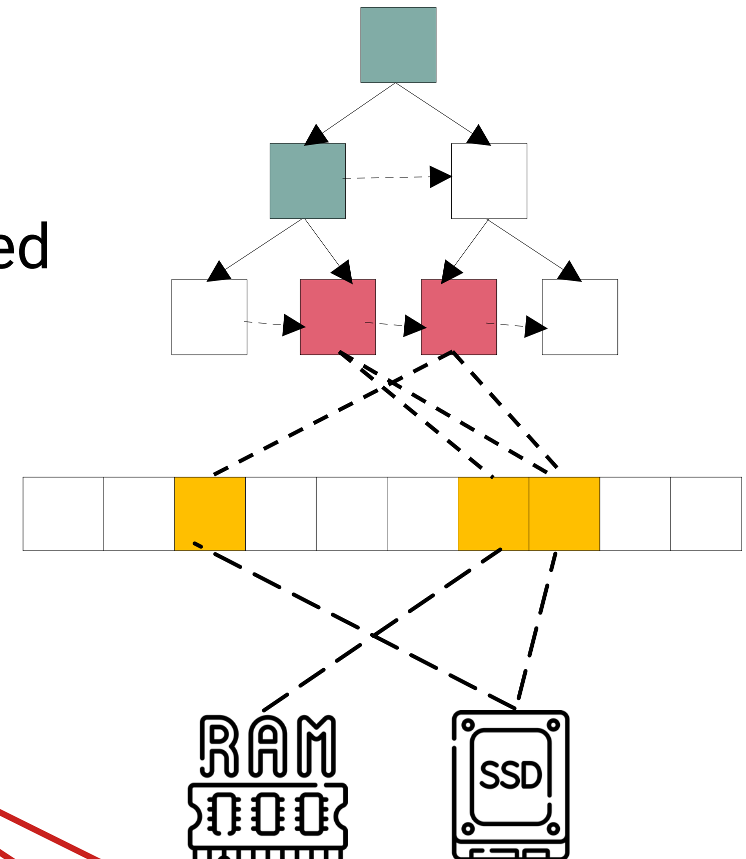
How to find them?

Example: Index scan

- Cost factors:
  - Inner tree blocks ~ log size of table
  - Tree leaf blocks ~ number of selected rows
  - Table blocks:
    - if clustered ~ <u>selected rows</u> / rows per block
    - if somewhat <u>clustered</u>… ?
    - if not clustered ~ selected rows
- Average cost of block depends on effectiveness of caching

How to estimate?

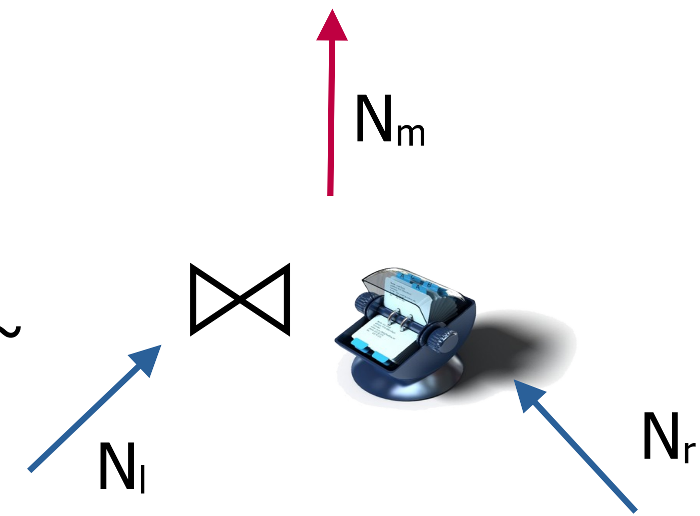
How to estimate?



Icons from Flaticon.

Example: Hash Join

- CPU cost:
  - Building hash table from inner table ~ N_r input rows
  - Checking each row in the outer table ~ N_l input rows
  - Creating the resulting <u>N_m rows</u>
- Very high cost if not enough memory to hold N_i rows in the hash table



How to estimate?

Coefficients

- Cost coefficients depend on the actual hardware, software, and even workloads

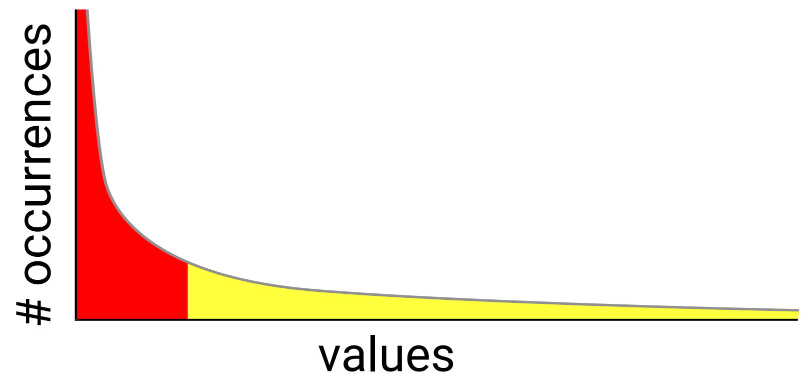
- Can be estimated by profiling simple workloads
- Can be <u>tuned by the DBA</u>

Cardinality - Selection

- Assumption of uniform distribution
- Know #distinct
  - Expected copies of each tuple:
    - #rows / #distinct

Cardinality - Selection

- Real data are usually not uniformly distributed:
  - 80/20 rule
  - Power law



- Know <u>most popular</u> and <u>#occurrences</u>
- Compute estimated #occurrences of others as uniformly distributed

 Image source: Wikipedia

Cardinality - Selection

- Data often have complex multimodal distributions
- Know histogram:
  - % of occurrences in each interval or
  - interval for fixed % of occurrences
- Compute #occurrences as uniformly distributed within each interval

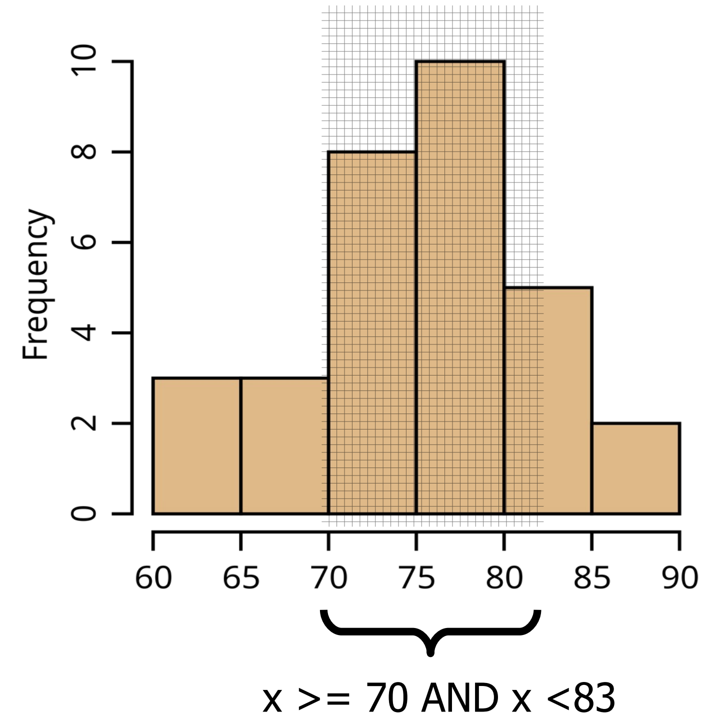


Image source: Wikipedia

Cardinality - Conjunction

- Filter conditions are often the conjunction of conditions on different columns
- Statistics on multiple columns are expensive:
  - Multidimensional
  - Many possible combinations
- Assume independently distributed values in different columns:
  - selectivity = #selected / #total rows
  - selectivity(a $\wedge$ b) = selectivity(a) * selectivity(b)
- Idem for disjunction

Cardinality - Join

- Cardinality for cross-product of A and B:
  - #rows from A × #rows from A
- Cardinality for join, first attempt:
  - Align buckets of histograms for A and B
  - Estimate the cardinality of each matching bucket as a cross-product:
    - Not good, as there are non-matching values
      - e.g. A has even numbers, B has odd numbers → no match!

Cardinality - Join

- Assume that one relation A has all values (containment)
- Cardinality for join, second attempt:
  - Align buckets of histograms for A and B
  - Estimate the cardinality of each matching bucket:
    - Each of #rows in B matches (#rows in A / #distinct in A)
      - Because A has all the values
    - Estimate as #rows B × #rows in A / #distinct in A
  - Generalize for the case where B has all values:
    - #rows in A × #rows in B / max(#distinct in A, #distinct in B)

Summary

- Tradeoff between complexity (data and computation) and accuracy
- Many other techniques:
  - More statistics
  - Heuristics
  - <u>Hinting</u>
  - Sampling of query
  - Machine learning
  - …

Cost model in PostgreSQL

- Computes two costs for each (sub-)plan:
  - (cost of first row .. cost of last row)
  - Interesting cases:
    - (0 .. big value) → row at a time operators
    - (big value … big value) → full relation operators
- Coefficients can be tuned in postgresql.conf
  - See “Planner Cost Constants”

Statistics in PostgreSQL

- Automatically keeps:
  - # distinct / # records
  - most popular
  - histogram
  - correlation
    - measures sorting / clustering of a column
- Computation of statistics:
  - Using sampling on large tables
  - Explicitly with ANALYZE
  - Implicitly as part of the “autovaccum” process

Statistics in PostgreSQL

- Multivariate statistics relate multiple columns:
  - Avoid the assumption of independence
- Statistics on multiple columns or arbitrary expressions
- Available statistics:
  - ndistinct → distinct combinations of columns / expressions
  - mcv → most common combinations
  - dependencies → relation between columns / expressions
- Useful for join estimation when containment assumption does not hold