Evaluation of Relational Operations

Yanlei Diao

Slides Courtesy of R. Ramakrishnan and J. Gehrke
Overview of Query Processing
Relational Operations

- We will consider how to implement
  - **Selection** (\(\sigma\)) Selects a subset of rows from relation.
  - **Join** (▷) Allows us to combine two relations.
  - **Projection** (\(\pi\)) Deletes unwanted columns from relation.
  - **Union** (\(\cup\)) Tuples in either reln. 1 or reln. 2.
  - **Intersection** (\(\cap\)) Tuples in both reln. 1 and reln. 2.
  - **Set-difference** (−) Tuples in reln. 1, but not in reln. 2.
  - **GROUP BY** and **Aggregation** (SUM, MIN, etc.)

with cost estimation, which leads to **cost-based optimization**
For more details, see the textbook

Database Management Systems
3rd Edition
Ramakrishnan and Gehrke

Amazon:
- Buy new: $43-$147.09 (hardcover); paperback, $23; Kindle, rent options are also available…
Outline

- Selection
- Sorting routine
- Join
- Projection
- Set operators
- Group By aggregation
Schema for Examples

Sailors (\textit{sid}: integer, \textit{sname}: string, \textit{rating}: integer, \textit{age}: real)
Reserves (\textit{sid}: integer, \textit{bid}: integer, \textit{day}: date, \textit{rname}: string)

- **Sailors:**
  - Each tuple is 50 bytes long,
  - 80 tuples per page,
  - 500 pages.

- **Reserves:**
  - Each tuple is 40 bytes long,
  - 100 tuples per page,
  - 1000 pages.

- **Cost metric:** \# I/Os
Using an Index for Selections

Cost of selection includes:
1) top down search in the index
2) scan the relevant leaf nodes
3) retrieve records from file (could be large w/o clustering)
Using an Index for Selections

- Cost of selection includes:
  1) top down search in the index
  2) scan the relevant leaf nodes
  3) retrieve records from file (could be large w/o clustering)

- Step 1) top down search: \( \leq 3-4 \) I/Os, depending on the height of the tree (and buffer management)
Cost Factors of Steps 2 and 3

- Cost of selection includes:
  1) top down search in the index
  2) scan the relevant leaf nodes
  3) retrieve records from file (could be large w/o clustering)

- Step 2 scanning leaf nodes: cost factors include
  - Reduction factor (% of qualifying tuples): rating > 8: 20% of tuples qualify
  - Num. of leaf nodes: If a data entry is 1/5 of a tuple, there are 500 / 5 = 100 leaf nodes.
  - So, 100 * 20% = 20 I/Os.

```
SELECT * 
FROM Sailors S 
WHERE S.rating > 8
```
Step 3 retrieving records from file: cost factors include

- **Num. of qualifying tuples**: rating > 8: 20% of tuples, 500*20%=100 data pages, 80*100=8,000 tuples
- **Clustering**: retrieving records from file
  - **Clustered index**: 100 I/Os.
  - **Unclustered index**: worst case 1 I/O per tuple; 8,000 I/Os here!
  - **Unclustered index + Sorting or Bitmap on rid**: ≤ 500 I/Os.
    (Bitmap Index Scan + Bitmap Heap Scan in PostgreSQL)
General Selections

- Boolean combination of predicates using AND and OR.
  - Conjunctive Normal Form (CNF), e.g.,
    
    \[
    \text{pred1 AND (pred3 OR pred4)}
    \]
    
    \[
    (\text{pred1 OR pred2}) \text{ AND (pred3 OR pred4)}
    \]

- *File scan* always works for general selections.

- *Index scan* works when it matches a predicate that is a conjunct of CNF.
  - E.g., an index matching ‘\text{pred1}’ can be used for
    
    \[
    \text{pred1 AND (pred3 OR pred4)}
    \]