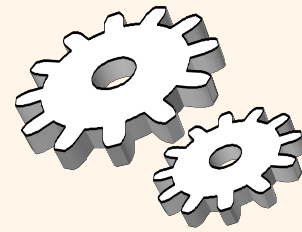
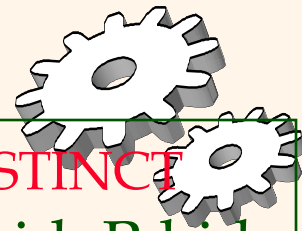


Evaluation of Relational Operations: Other Techniques



Simple Selections

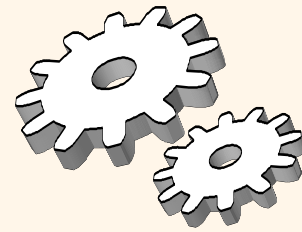
- ❖ $\text{Select (R.attr op value) R}$
- ❖ No index, unsorted: must scan whole table
- ❖ With an index on selection attribute: use it



The Projection Operation

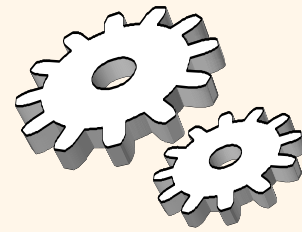
```
SELECT  DISTINCT  
        R.sid, R.bid  
FROM    Reserves R
```

- ❖ An approach based on sorting:
 - Modify Pass 0 of external sort to eliminate unwanted fields.
 - Modify merging passes to eliminate duplicates.



Projection Based on Hashing

- ❖ *Partitioning phase*: Read R using one input buffer. For each tuple, discard unwanted fields, apply hash function $h1$ to choose one of $B-1$ output buffers.
 - Result is $B-1$ partitions (of tuples with no unwanted fields).
2 tuples from different partitions guaranteed to be distinct.
- ❖ *Duplicate elimination phase*: For each partition, read it and build an in-memory hash table, using hash fn $h2$ ($\neq h1$) on all fields, while discarding duplicates.
 - If partition does not fit in memory, can apply hash-based projection algorithm recursively to this partition.



Set Operations

- ❖ Intersection and cross-product special cases of join.
- ❖ Union (Distinct) and Except similar; we'll do union.
- ❖ Sorting based approach to union:
 - Sort both relations (on combination of all attributes).
 - Scan sorted relations and merge them
- ❖ Hash based approach to union:
 - Partition R and S using hash function h .
 - For each S-partition, build in-memory hash table (using h_2), scan corr. R-partition and add tuples to table while discarding duplicates.

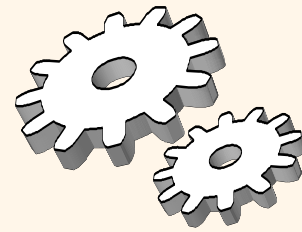
Aggregate Operations (AVG, MIN, *etc.*)

❖ Without grouping:

- In general, requires scanning the relation.
- Given index whose search key includes all attributes in the SELECT or WHERE clauses, can do index-only scan.

❖ With grouping:

- Sort on group-by attributes, then scan relation and compute aggregate for each group. (Can improve upon this by combining sorting and aggregate computation.)
- Similar approach based on hashing on group-by attributes.
- Given tree index whose search key includes all attributes in SELECT, WHERE and GROUP BY clauses, can do index-only scan



Summary

- ❖ A virtue of relational DBMSs: *queries are composed of a few basic operators*; the implementation of these operators can be carefully tuned (and it is important to do this!).
- ❖ Many alternative implementation techniques for each operator; no universally superior technique for most operators.
- ❖ Must consider available alternatives for each operation in a query and choose best one based on system statistics, etc. This is part of the broader task of optimizing a query composed of several ops.