

## **HEURISTICS for choosing indexes (from Connolly & Begg)**

1. Do not index small relations. May be better to search.
2. Index the primary key of a relation.
3. Add secondary index to a foreign key, if frequently accessed.
4. Add secondary index to any attribute that is heavily used as a secondary key.
5. Add a secondary index on attributes that are frequently involved in:
  - (a) selection criteria
  - (b) join criteria
  - (c) ORDER BY
  - (d) GROUP BY
  - (e) other operations involving sorting, such as UNION, DISTINCT
  - (f) built-in aggregate functions (e.g. SUM(salary) )
6. Avoid indexes on attributes or relations that are frequently updated.
7. Avoid indexing an attribute if the query will retrieve a significant proportion (e.g. 25%) of the tuples in the relation – it may be more efficient to search the entire relation.
8. Avoid indexing attributes that consist of long character strings.

Do not include indexes that negatively impact performance – e.g., a frequently updated relation with require its index be frequently updated.

If many tuples are being inserted into a relation with indexes, it may be more efficient to temporarily drop the indexes before doing the inserts.