08 – Tree Indexes 2

1. More B+Trees
    A. Duplicate Keys
        i. Add record id as key
        ii. Use overflow leaf nodes to contain duplicate keys
        maintain and modify issue
2. Index Technique
    A. Implicit Indexes – DBMS maintains indexes for integrity (constraint or etc)
    B. Partial Indexes –
    make indexes for subset of DB to make query more efficient
    C. Covering Indexes –
    if index has all attributes that are needed, than we don't need to do disk I/O.
    D. Index Include Columns -
    build index with some attributes to make query more efficient
    this attributes won't be used in record search
    E. Functional Indexes – do some operation on keys when building indexes
3. Tries / Radix Trees
    A. Trie == prefix tree
        i. All operation is O(k), (k == length of key)
        ii. Fan-out == # of bits that each key/digit represents
    B. Radix Tree == Patricia Tree
        i. Omit all nodes with only a single child
4. Inverted Indexes
    A. Tree indexes are useful for "point search" and "range search",
    but, these are not useful for "keyword search".
    B. Map words to records that contain those words in the target attributes
5. What is it?
    A. How Inverted indexes work