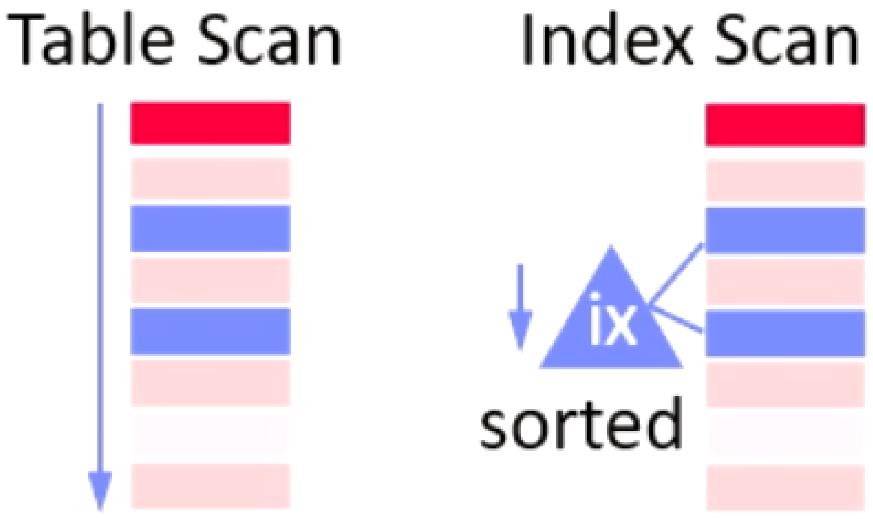
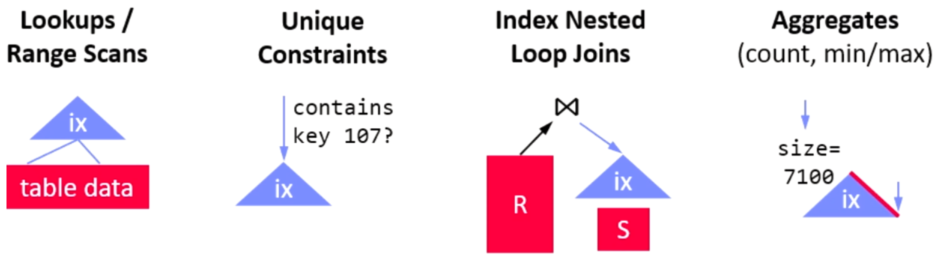
Table Scan vs Index Scan

- for highly selective predicates
  - index scan asymptotically way better than table scan
- index scan higher per tuple overhead
  - break even at ~5% output ratio
  - index scan better if filter ratio below ~5%
- multi-column predicates
  - fetch/RID-list intersection
- 

Use Cases for Indexes



Create Index

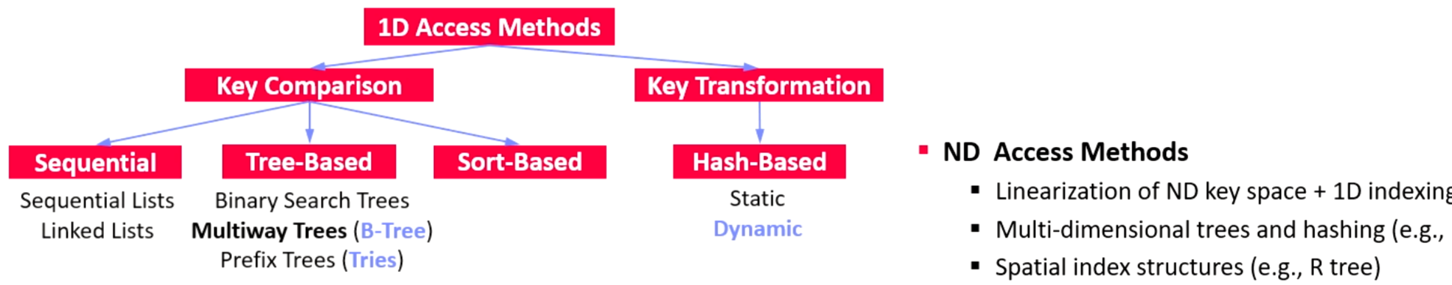
- create secondary (nonclustered) index on set of attributes
  - clustered: tuples sorted by index
  - nonclustered: sorted attribute with tuple references

```
CREATE INDEX ixStudLname
  ON Students USING btree
  (Lname ASC NULLS FIRST);

DROP INDEX ixStudLname;
```

-
- allows specifying uniqueness, order, indexing method
- postgreSQL methods
  - [[Binary Search and BTree]]
  - hash
  - gist
  - gin

Classification of Index Structures



- **ND Access Methods**
  - Linearization of ND key space + 1D indexing
  - Multi-dimensional trees and hashing (e.g.,
  - Spatial index structures (e.g., R tree)