## Chapter 4.6 Algorithm based on Index

***Instruction:***

Several kinds of Algorithm need Index on one or more fields in Relation. Selection Operation based on Index is especially useful, but for Join Algorithm and other Binary Operator Algorithm also have some benefits. In this chapter, we will introduce the algorithm.

### Chapter 4.6.1 Cluster and Non - Cluster Index

***Definition:***

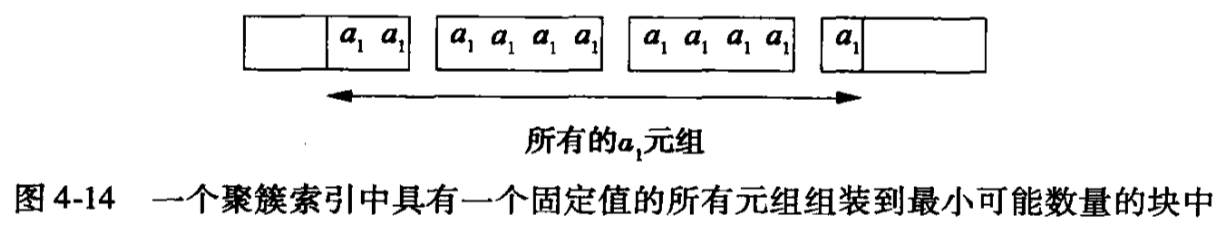
1. Cluster - If the number of tuple in one Relation can be contracted to store as little as possible, then this Relation is Cluster. All analysis so far are all based on cluster.
2. Cluster Index on one or more attributes, then all tuples with the fixed value can be put into little blocks.

***Attention:***

Non - cluster Relation can not have one cluster index, but one Cluster Relation can have non - cluster index.

***Example:***

* Relation R(a, b), sorted by property a and stored based on this sequence, then if Relation R(a, b) stores into the block, it must be cluster. For one given property *a* - with value *a1*, all tuples with value *a1* are stored sequentially, so the index on property *a* is cluster index.
* See the picture below, except the first and the last block which contain the value *a1*, they are all stored on the block. But actually, index on the property *b* may not be cluster, since there has one tuple with the fixed value *b* can be divided into the file, except that there has close relationship with property *a* and *b*.



### Chapter 4.6.2 Selection based on Index

### Chapter 4.6.3 Connection by Using Index

### Chapter 4.6.4 Join by Using Ordered Index