# Chapter 3.5 Hash Structure of Multi-Dimensional Data

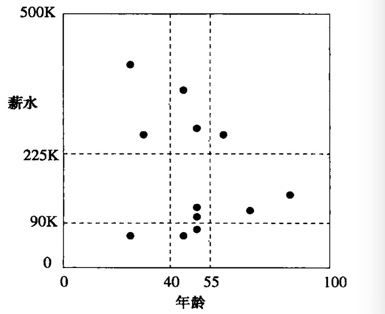
Build hash function using single key gets another two data structures:

* Grid File:Sort values of the dimension to divide this dimension.
* Piecewise Hash Function: Hash each dimensions and every dimension does influence the bucket.

## Chapter 3.5.1 Grid File

The simplest data structure compared with single index (hash function with single key) is Grind File. (The space be divided as the grind line space).

***Principles:***

* Grind line separates the dimension into strip, the dots on the grind line will be considered belongs to the strip which is lower than this grind line.
* Different grind can have different numbers of grind line. Interval length can be different within the neighboring grind lines.
* The same grind line can have different interval length.

***Example:***

Insert customer with two properties (age, salary) into two-dimension space:

|  |  |  |  |
| --- | --- | --- | --- |
| (25, 60) | (45, 60) | (50, 75) | (50, 100) |
| (50, 120) | (70, 110) | (85, 140) | (30, 260) |

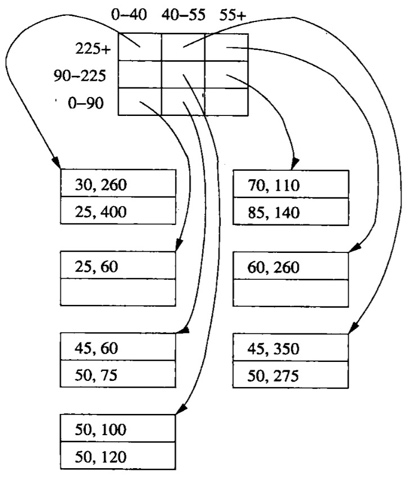
Look from the right image, twelve points are located in a single two-dimensional space. In every single dimensional, two lines are used to divide the current dimension into three parts. In sum, there are nine spaces in the current two-dimension space.

* 225K <= salary <= 500K, 0 <= age < 40;
* 225K <= salary <= 500K, 40 <= age < 55;
* 225K <= salary <= 500K, 55 <= age <= 100;
* 90K <= salary < 225K, 0 <= age < 40;
* 90K <= salary < 225K, 40 <= age < 55;
* 90K <= salary < 225K, 55 <= age <= 100;
* 0K <= salary < 90K, 0 <= age < 40;
* 0K <= salary < 90K, 40 <= age < 55;
* 0K <= salary < 90K, 55 <= age < 100;

## Chapter 3.5.2 Searching on Grid File

***Principles:***

* See every space as a bucket of hash table, and every single record in each space is seen as the record of this bucket.
* In order to locate a record, the value of every dimension should be confirmed. We focus on every record and its location in the grid space.
* The location of every single dimension decides its location in its bucket.

***Example:***

The bucket here is a 3\*3 matrix which is used to indicate nine spaces.

* Two empty spaces.
* Two records at most in every single bucket.
* Overflow blocks are un-necessary if no bucket has exceeded two records.

## Chapter 3.5.3 Insertion in Grid File

***Description:***

When record is needed to be inserted into a grid file, then we search the bucket and put the record into the bucket.

1. Do not find space in the bucket.

***Principle:***

* Find space in the bucket. -> Nothing has to be done.
* Can not find space in the bucket.
  + Add overflow blocks to the bucket.

***Example:***

## Chapter 3.5.4 Grid File Performance

## Chapter 3.5.5 Piecewise Hash Function

## Chapter 3.5.6 Compare Between Grid File and Piecewise Hash Function