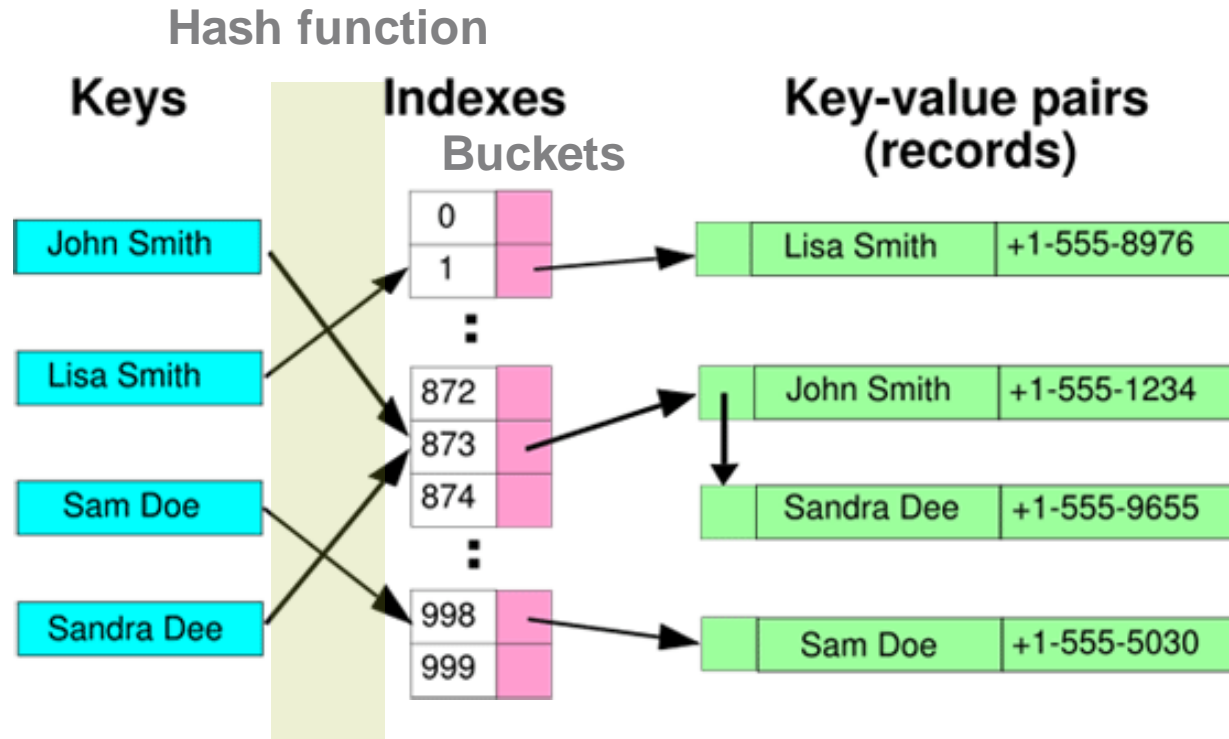


해쉬 테이블

김수진

해쉬 테이블



John smith의 Key값을 Hash 함수를 통해 873으로 변환하여 접근
탐색할 때 상수 시간 $O(1)$ 이 걸린다.

해쉬 테이블

- Key와 value 쌍을 저장하는 자료구조
- DB의 기초
- 검색이 매우 빠름
- 순서를 예측 X

Hash 값이 충돌할 수도 있음 -> 같은 해쉬 값끼리 리스트로 저장

데이터들이 한곳에 모이는 문제 발생 -> 클러스터

해쉬 함수 알고리즘

1. 나눗셈 법 :

해쉬 값이 테이블의 크기를 넘는 것을 방지하기 위해 입력 값을 테이블의 크기로 나누고 나머지를 테이블의 주소로 사용한다.

$$\text{주소} = \text{입력 값} \% \text{테이블의 크기}$$

2. 자릿 수 접기 :

숫자를 일정한 크기 이하로 만드는 방법

$$\text{ex) } 8129335 = 8 + 1 + 2 + 9 + 3 + 3 + 5 = 31$$

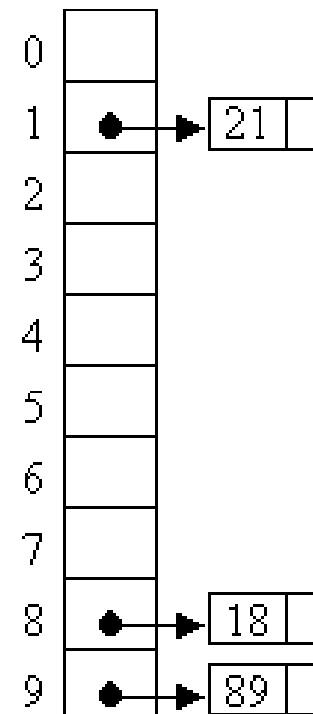
$$81 + 29 + 33 + 5 = 148$$

해쉬 충돌

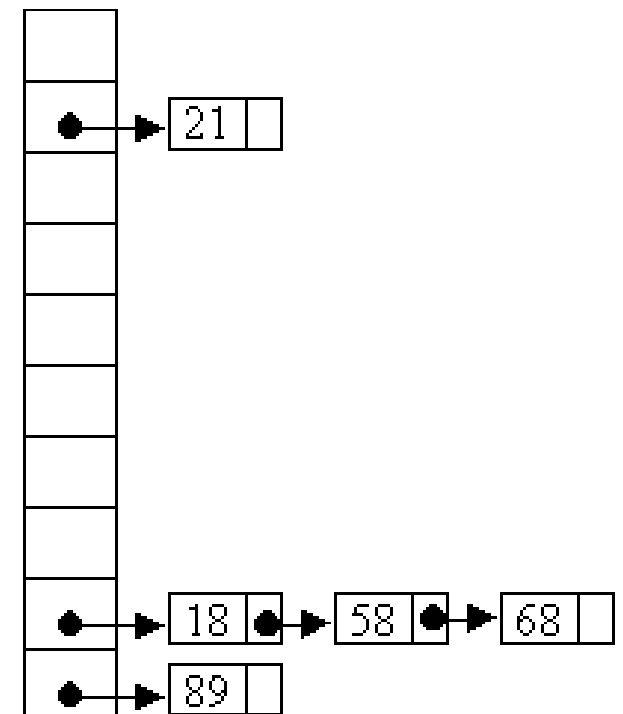
개방 해싱 알고리즘 :

충돌이 발생하면 각 데이터를 해당 주소에 있는 링크드 리스트에 삽입하여 문제를 해결하는 방법.

Insert
18, 89, 21



Insert
58, 68

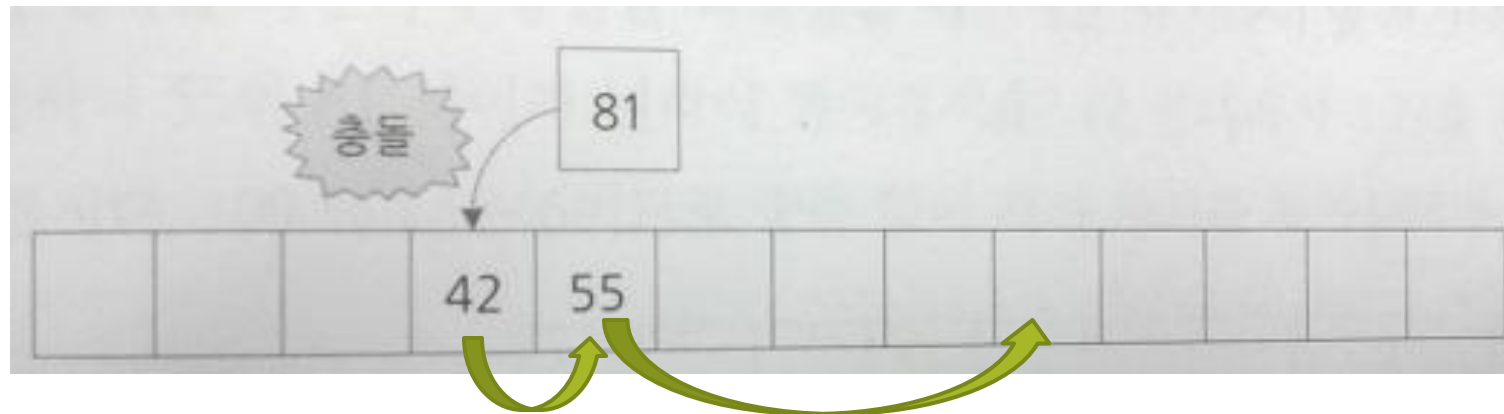


해시 충돌

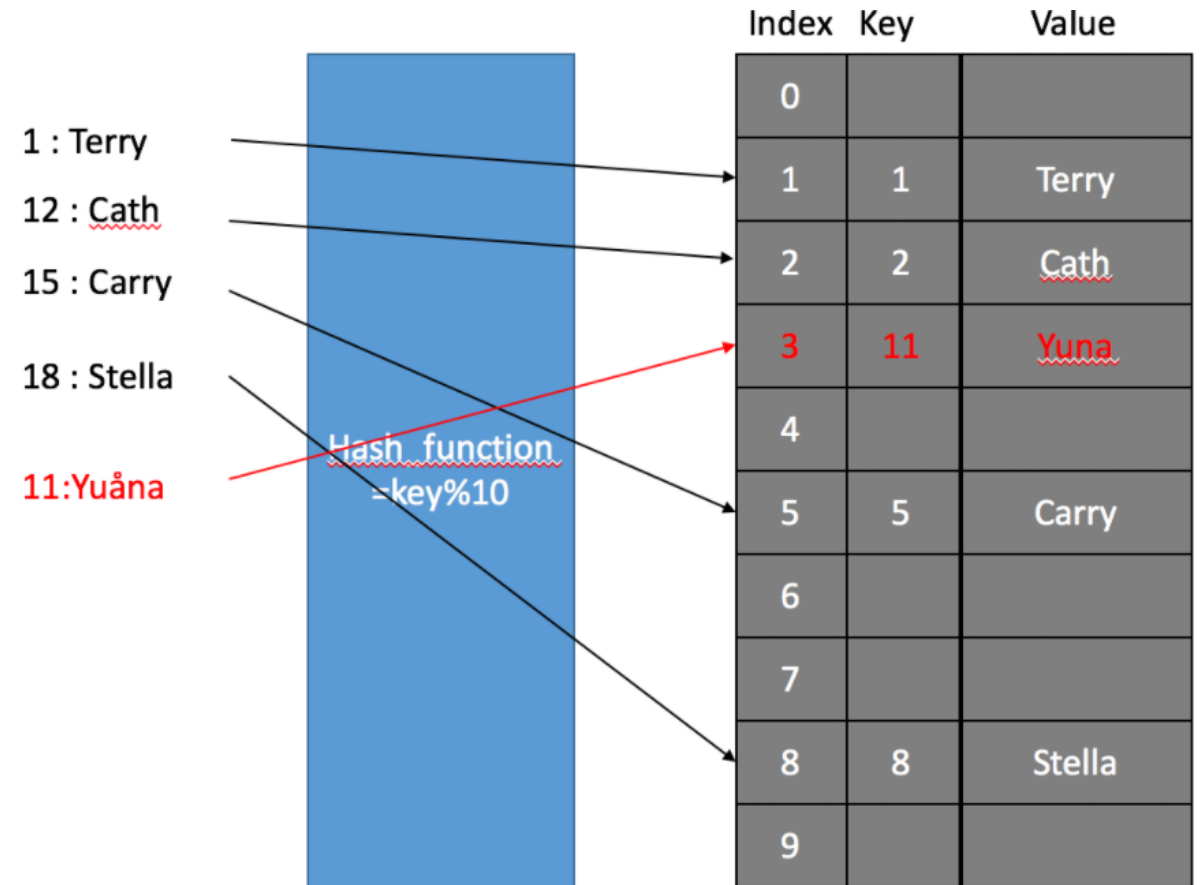
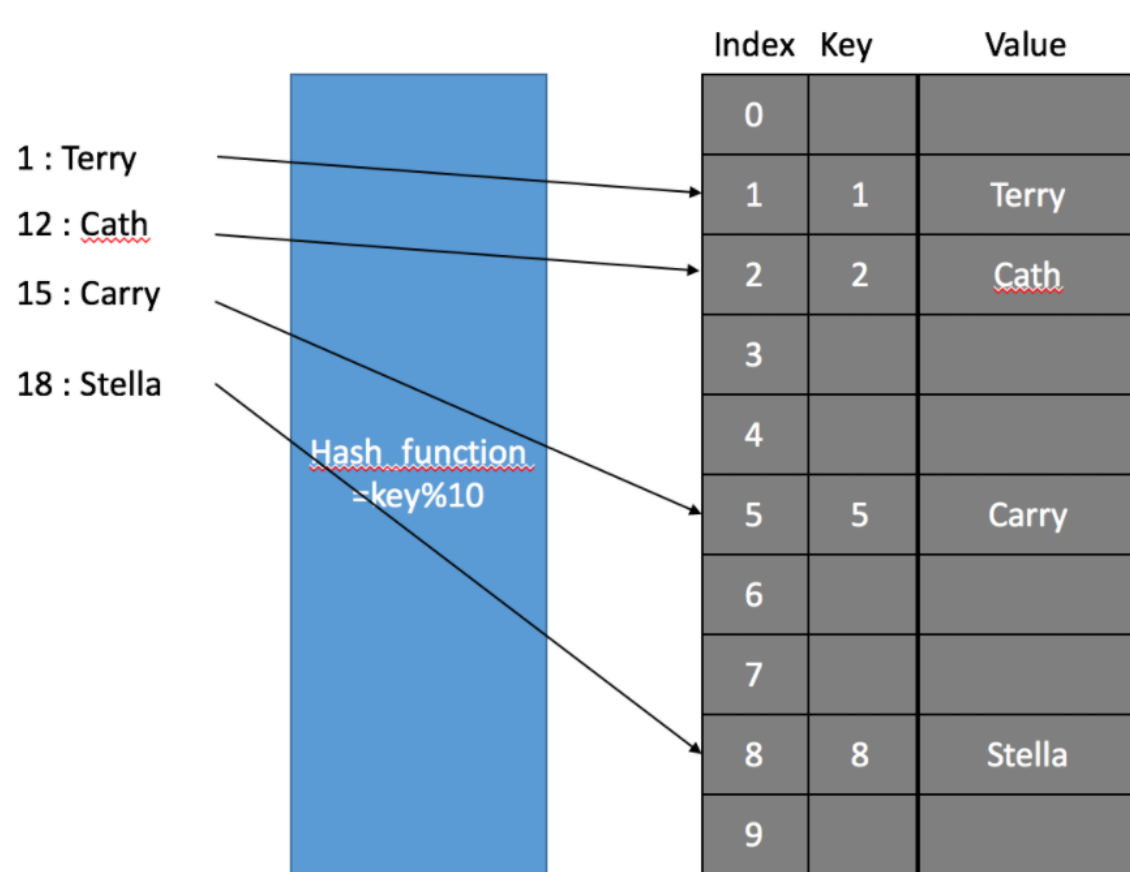
개방 주소 법: 클러스터 발생

충돌이 일어나면 해시 테이블 내의 새로운 주소를 탐사하여 충돌된 데이터를 입력하는 방식

- 1) 선형 탐사 (고정 폭(ex 1)으로 다음 주소로 이동하여 넣는다)
- 2) 제곱 탐사 (이동 폭이 제곱수로 늘어남)
- 3) 이중 해싱(2개 해시 함수), 재 해싱(크기를 늘려 다시 해싱하는 것)



선형 탐사 예 (삽입)



선형 탐색 예 (삭제)

Index	Key	Value
0		
1	1	Terry
2	2	Cathy
3	11	Yuna
< 4		
5	5	Carry
6		
7		
8	8	Stella
9		

Index	Key	Value
0		
1	1	Terry
Dummy node		
2	11	Yuna
3		
4		
5	5	Carry
6		
7		
8	8	Stella
9		