

제 1 장

파일 처리의 개념



SQL Server

ORACLE®

목 차

1.1 개요

1.1.1 자료의 표현 단위

1.1.2 파일의 기본 구성

1.1.3 파일의 계층 구조

1.2 파일의 분류

1.2.1 사용 매체에 따른 분류

1.2.2 내용에 따른 분류

1.2.3 파일 구조에 따른 분류

1.2.4 파일의 연산

1.1 개요

- 자료의 표현 단위

- 데이터 : 프로그램 수행에 필요한 자료

- 비트(bit)

- 2진법 한 자리를 기억하는 자료 표현의 최소 단위

- 바이트(Byte)

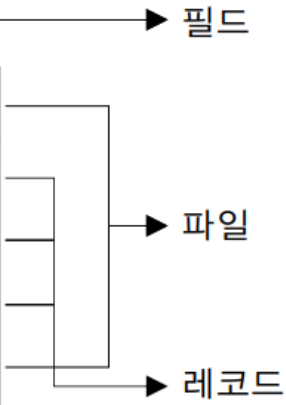
- 하나의 문자 또는 숫자를 표현하는 최소 단위로 8bit가 모여 구성되는 비트의 집합
코드(ASCII 코드, EBCDIC 코드)

1.1.2 파일의 기본 구성

- 필드(Field, Item)

- 특정 의미를 나타내기 위해서 하나 이상의 바이트로 구성됨
- 프로그램의 데이터 처리 단위
- 레코드를 구성하는 기본 구성 단위

학번	성명	성별	생년월일	우편번호	주소
2011001	이영선	여	94.06.04	135-142	부산
2008032	김미숙	여	92.11.30	321-840	충남
2010003	박선병	남	89.06.13	401-240	인천
2009021	최수영	남	90.01.17	664-010	경남



1.1.2 파일의 기본 구성

● 레코드(Record)

- 여러 개의 항목(field)이 모여서 하나의 단위로 취급될 수 있는 **연관성**을 가진 데이터
- 파일의 구성 요소
- 특정 의미를 나타내기 위해서 하나 이상의 바이트로 구성됨
- 논리적 레코드, 물리적 레코드(블록, block)
- 물리적 레코드 : 파일 처리에서 입출력되는 최소 단위

● 파일(File)

- 정보 처리의 대상으로 연관된 레코드의 집합

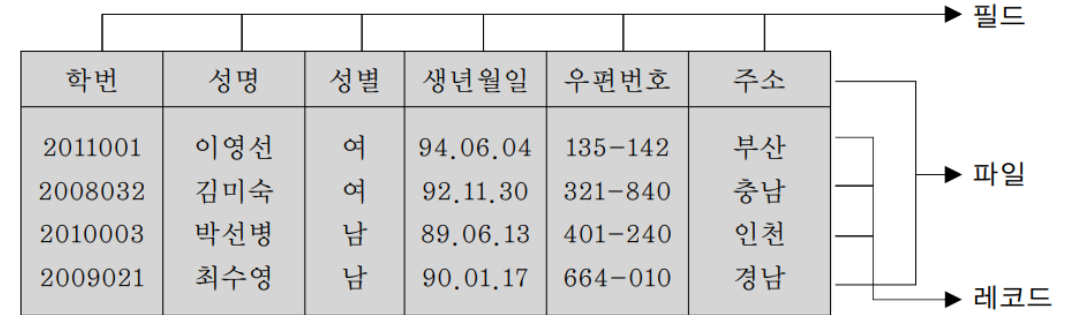


그림 1.2 주소록 파일

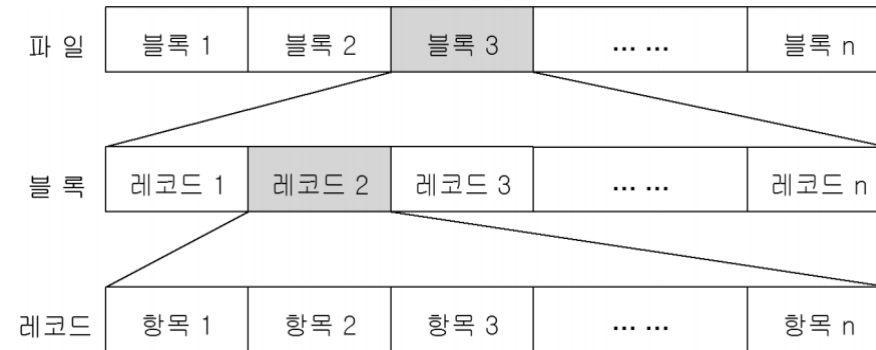
1.1.2 파일의 기본 구성

학번	성명	성별	생년월일	우편번호	주소
2011001	이영선	여	94.06.04	135-142	부산
2008032	김미숙	여	92.11.30	321-840	충남
2010003	박선병	남	89.06.13	401-240	인천
2009021	최수영	남	90.01.17	664-010	경남

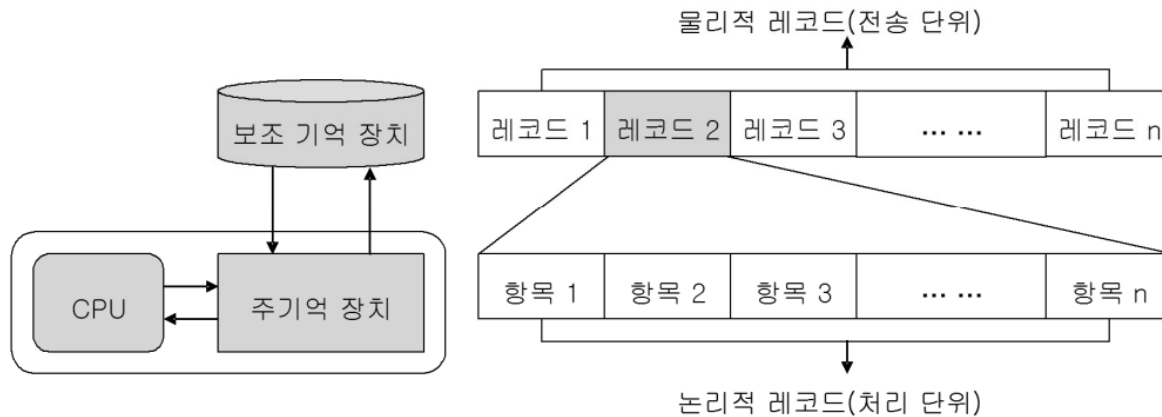
비트 ∈ 바이트 ∈ 필드 ∈ 레코드 ∈ 파일 ∈ 데이터베이스

1.1.3 파일의 계층 구조

- 파일은 여러 개의 레코드로 구성
- 레코드는 여러 개의 필드(항목)으로 구성
- 물리적 레코드(블록)
 - 주기억 장치와 보조기억장치와의 데이터 전송 단위
 - 블록에는 여러 개의 논리적 레코드를 포함



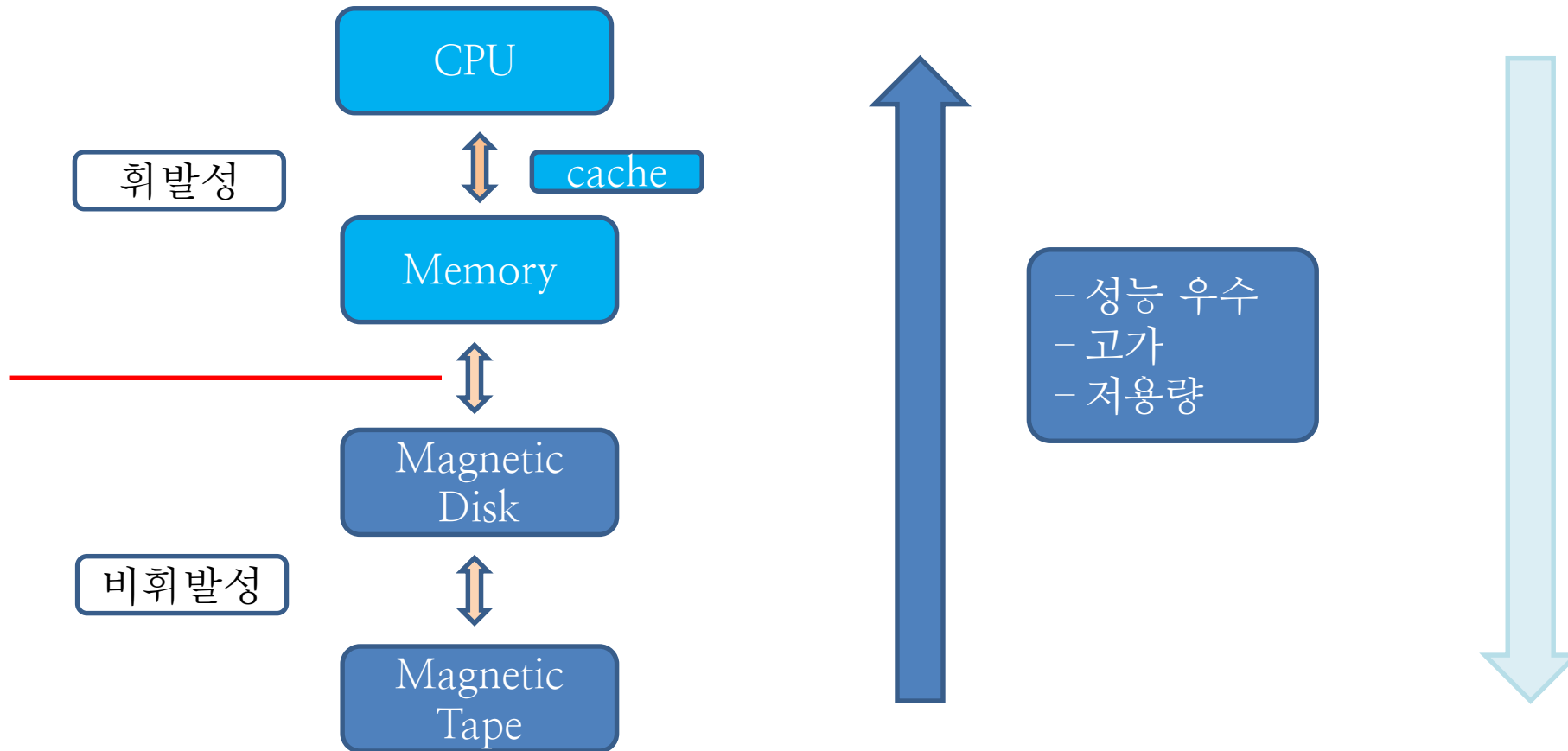
파일의 계층 구조



물리적 레코드와 논리적 레코드의 관계

1.2.1 파일의 저장 매체에 대한 분류

● 메모리의 계층 구조



1.2 파일의 분류

- 기준에 따른 파일의 분류
 - 파일 저장 매체에 따른 분류
 - 파일 내용에 따른 분류
 - 파일 구조에 따른 분류

1.2 파일의 분류

- 기준에 따른 파일의 분류
 - 파일 저장 매체에 따른 분류
 - 파일 내용에 따른 분류
 - 파일 구조에 따른 분류

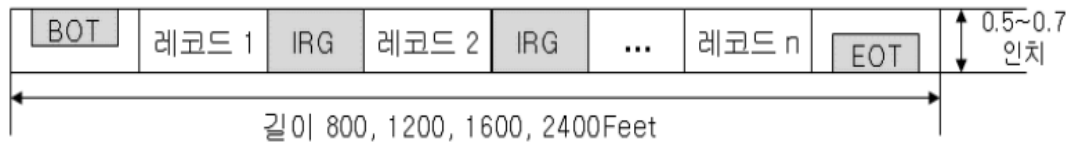
1.2.1 파일의 저장 매체에 대한 분류



1.2.1 파일의 저장 매체에 대한 분류

● 자기 테이프(MT; Magnetic Tape) 파일

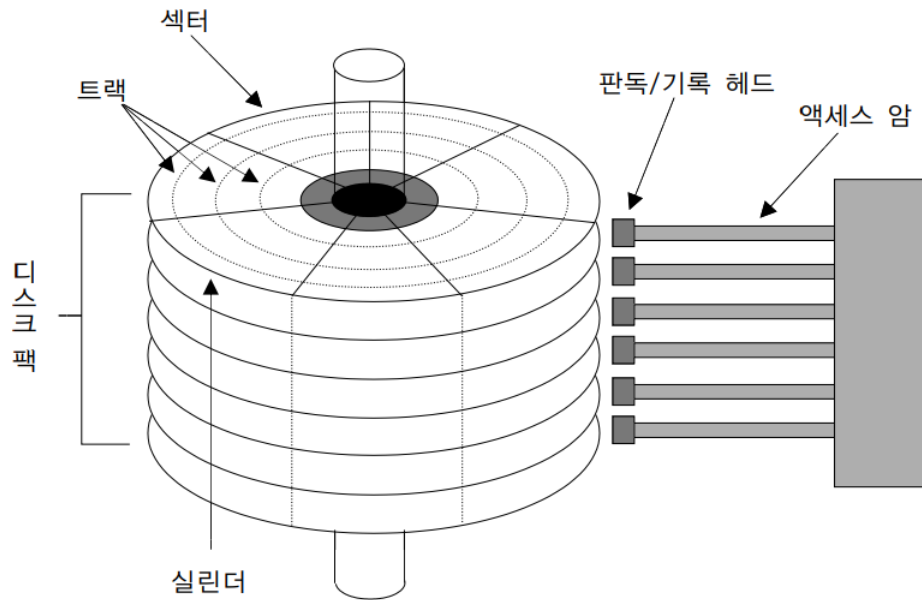
- 순차 접근 저장 장치(SASD; Sequential Access Storage Device)
- IRG(Inter Record Gap) / IBG(Inter Block Gap)
- BOT(Begin Of Tape) / EOT(End Of Tape)
- BPI(byte Per Inch)



1.2.1 파일의 저장 매체에 대한 분류

● 자기 디스크(Magnetic Disk) 파일

- 직접 접근 저장 장치(DASD; Direct Access Storage Device)
- Sector / Track / Cylinder
- 이동형 디스크 헤드 / 고정형 디스크 헤드



자기 디스크의 구성



1.2 파일의 분류

- 일괄 처리(batch mode)와 대화식 처리(interactive mode)

1.2.2 파일 내용에 따른 분류

데이터 파일(data file)	마스터 파일(master file)
	트랜잭션 파일(transaction file)
	히스토리 파일(history file)
	보고서 파일(report file)
프로그램 파일(program file)	제어프로그램 파일(control program file)
	처리프로그램 파일(process program file)
워크 파일(work file)	작업 파일
	입력데이터 파일
	출력데이터 파일
	중간 파일
	체크포인트 파일

1.2.2 파일 내용(역할)에 따른 분류

- 마스터 파일(master file)
 - 일반 업무처리에서 처럼 원장이나 대장과 같이 시스템의 중심이 되는 파일
- 트랜잭션 파일(transaction file)
 - 마스터 파일에 대한 수정 데이터를 저장하고 있는 파일
- 요약 파일(summary file)
 - 업무를 편리하게 하기 위해 어떤 파일의 내용을 요약해서 만든 파일
- 히스토리 파일(history file)
 - 통계 처리나 파일의 자료에 잘못이 발생했을대 파일의 복구를 위해 사용되는 파일
- 보고서 파일(report file)
 - 각종 자료를 일정한 형식을 갖추어 출력하기 위한 파일(report writer)

1.2.3 파일 구조에 따른 분류

- 순차 파일(Sequential File)
 - 색인 순차 파일(Indexed Sequential File)
 - 랜덤 파일(Random File)
 - 링 파일(Ring File)
-
- SAM file(Sequential Access Method file)
 - ISAM file(Indexed Sequential Access Method file)
 - DAM file(Direct Access file)

1.2.3 파일 구조에 따른 분류

● 순차 파일(Sequential File)

- 레코드를 논리적 순서에 따라 저장된 파일
- 자기 테이프, 자기 디스크에 저장
- 정렬된 파일(sorted file) → sort key [field]

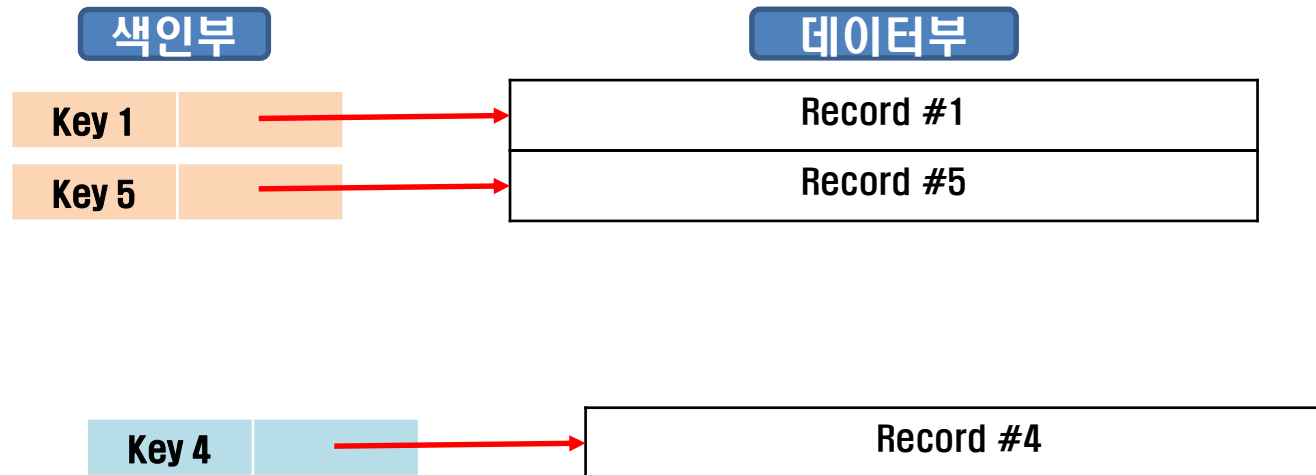
Record 1
Record 2
Record 3
Record 4
Record 5
.....

.....	데이터1	데이터2	데이터3	데이터4	데이터5	데이터6	데이터7	..
-------	------	------	------	------	------	------	------	----

1.2.3 파일 구조에 따른 분류

● 색인 순차 파일(Indexed Sequential File)

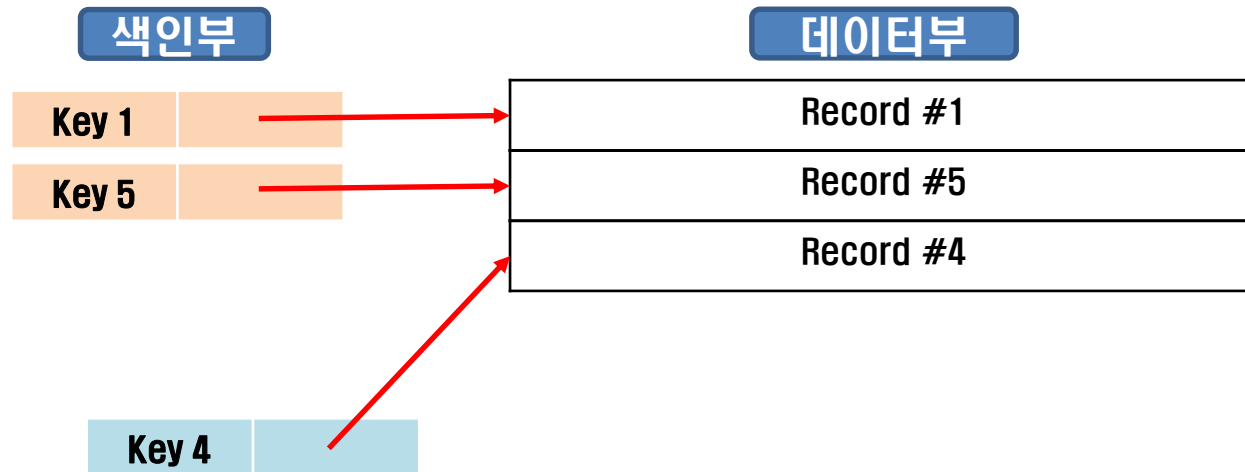
- 키 값에 따라 순서적으로 정렬된 데이터 부분과 데이터 부분에 대한 포인터를 가지고 있는 색인 부분으로 구성됨
- 키 값에 의한 순차적 접근과 임의 접근이 가능함.
- 색인 부분 : 빠른 검색을 목적으로 함, 기억 공간을 최소화하고, 검색 시간은 짧아지도록 설계함.



1.2.3 파일 구조에 따른 분류

● 색인 순차 파일(Indexed Sequential File)

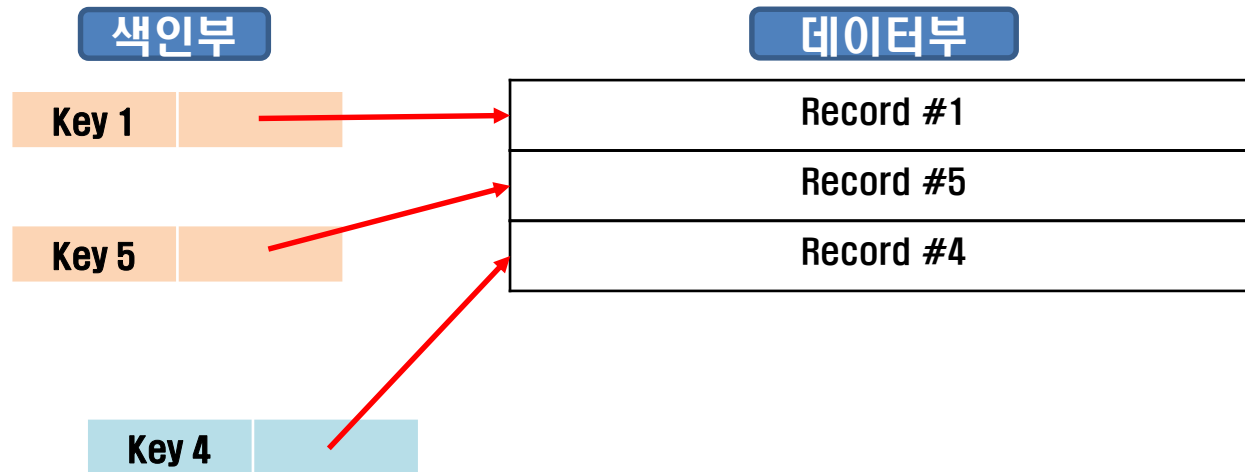
- 키 값에 따라 순서적으로 정렬된 데이터 부분과 데이터 부분에 대한 포인터를 가지고 있는 색인 부분으로 구성됨
- 키 값에 의한 순차적 접근과 임의 접근이 가능함.
- 색인 부분 : 빠른 검색을 목적으로 함, 기억 공간을 최소화하고, 검색 시간은 짧아지도록 설계함.



1.2.3 파일 구조에 따른 분류

● 색인 순차 파일(Indexed Sequential File)

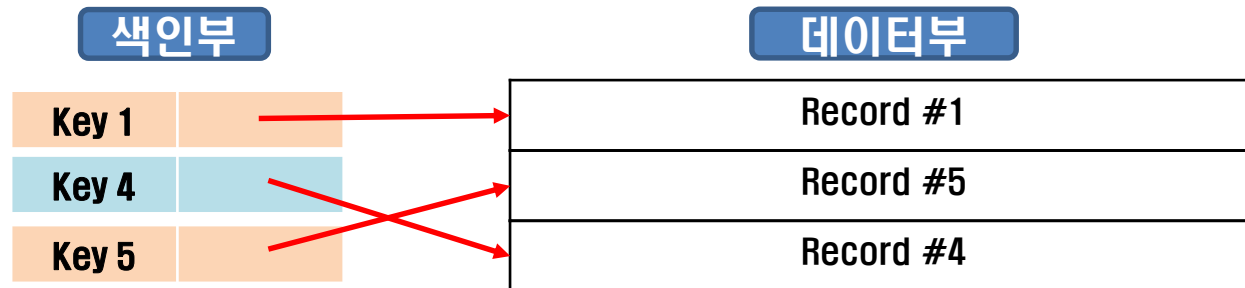
- 키 값에 따라 순서적으로 정렬된 데이터 부분과 데이터 부분에 대한 포인터를 가지고 있는 색인 부분으로 구성됨
- 키 값에 의한 순차적 접근과 임의 접근이 가능함.
- 색인 부분 : 빠른 검색을 목적으로 함, 기억 공간을 최소화하고, 검색 시간은 짧아지도록 설계함.



1.2.3 파일 구조에 따른 분류

● 색인 순차 파일(Indexed Sequential File)

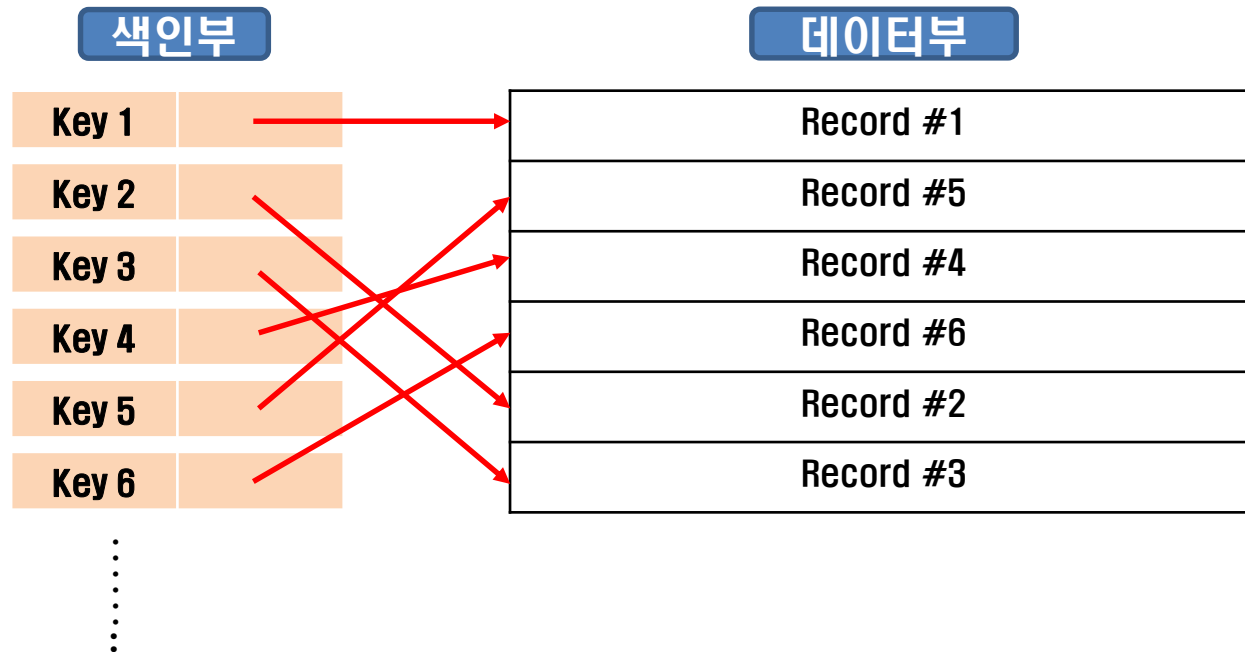
- 키 값에 따라 순서적으로 정렬된 데이터 부분과 데이터 부분에 대한 포인터를 가지고 있는 색인 부분으로 구성됨
- 키 값에 의한 순차적 접근과 임의 접근이 가능함.
- 색인 부분 : 빠른 검색을 목적으로 함, 기억 공간을 최소화하고, 검색 시간은 짧아지도록 설계함.



1.2.3 파일 구조에 따른 분류

● 색인 순차 파일(Indexed Sequential File)

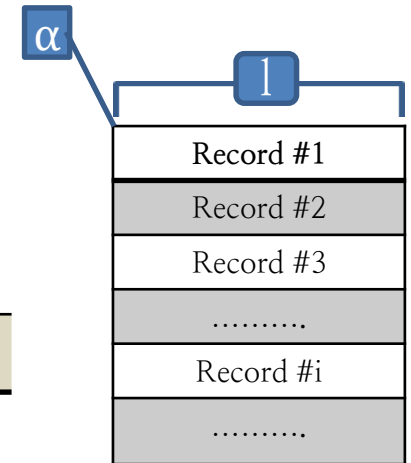
- 키 값에 따라 순서적으로 정렬된 데이터 부분과 데이터 부분에 대한 포인터를 가지고 있는 색인 부분으로 구성됨
- 키 값에 의한 순차적 접근과 임의 접근이 가능함.
- 색인 부분 : 빠른 검색을 목적으로 함, 기억 공간을 최소화하고, 검색 시간은 짧아지도록 설계함.



1.2.3 파일 구조에 따른 분류

● 직접 파일(Direct File)

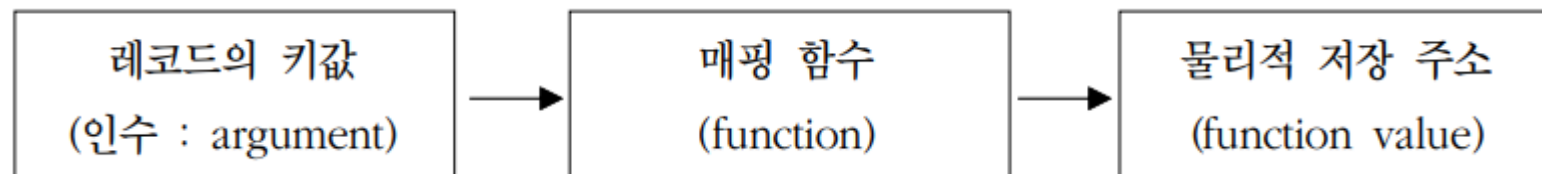
- 색인(index)부 없이 직접 접근이 가능함.
- 키를 이용해 해당 레코드의 주소를 계산함.



$$\text{addr}(\text{key}=1) = \alpha$$

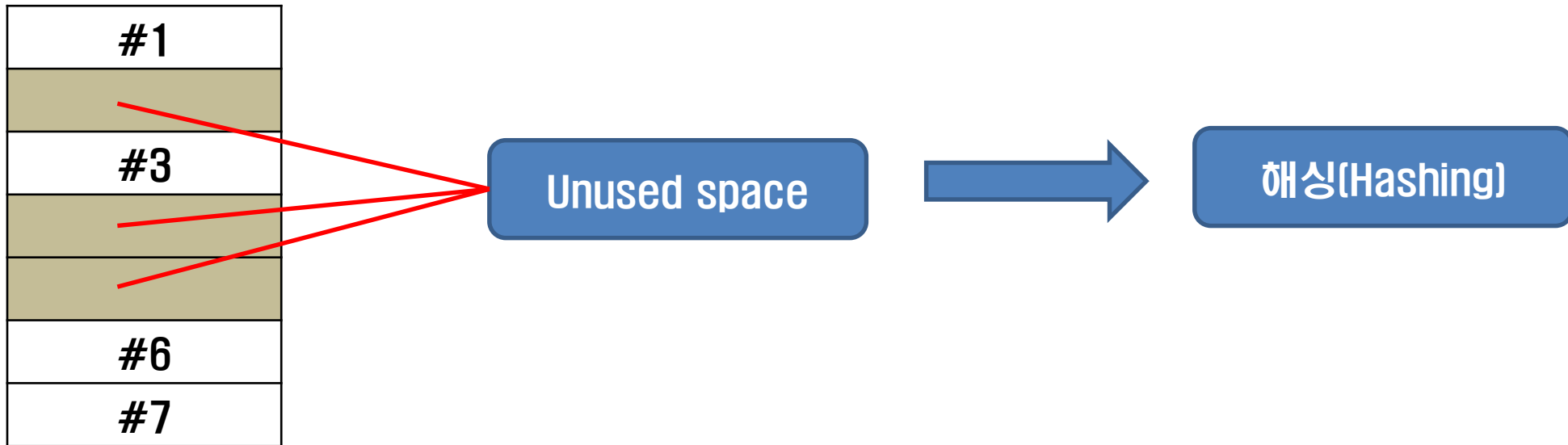
$$\text{addr}(\text{key}=2) = \alpha + 1$$

$$\text{addr}(\text{key}=i) = \alpha + (i-1)*1$$



1.2.3 파일 구조에 따른 분류

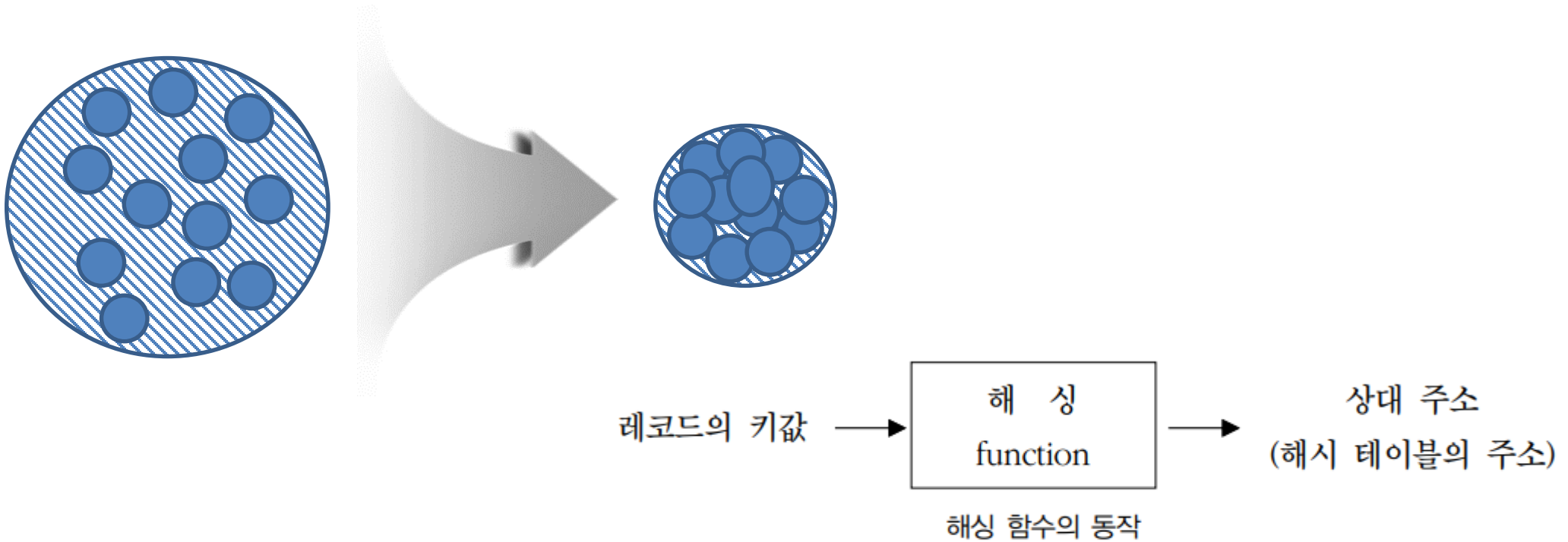
- 직접 파일(Direct File)의 문제점
 - 키값이 연속이지 않다면, 메모리를 낭비함.



1.2.3 파일 구조에 따른 분류

- 해싱(Hashing)

➤ 큰 영역의 메모리를 상대적으로 작은 범위를 갖는 상대주소 값으로 변경하는 함수를 이용하는 방법



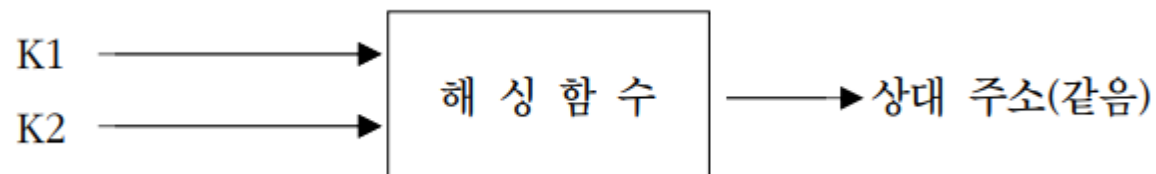
1.2.3 파일 구조에 따른 분류

● 해싱 함수(Hashing Function)

- 키는 숫자 2자를 이용함(00~99). -> 실제 데이터는 9개만 존재함
- 키를 10을 나누어 나머지를 상대주소로 이용함.

키	함수	상대주소
9	9/10 9	9
11	11/10 1	1
25	25/10 5	5
37	37/10 7	7
26	26/10 6	6
71	71/10 1	1
.....		

* 충돌현상(collision)



※ K1, K2는 동거자(synonyms)

1.2.3 파일 구조에 따른 분류

- 해싱 함수의 종류

- 중간 제공 방법
- 제산법
- 이동법
- 접는 방법
- 계수 분석법
- 진법 변화 방법
- 다항식의 제산법
- Exclusive-OR 변환법

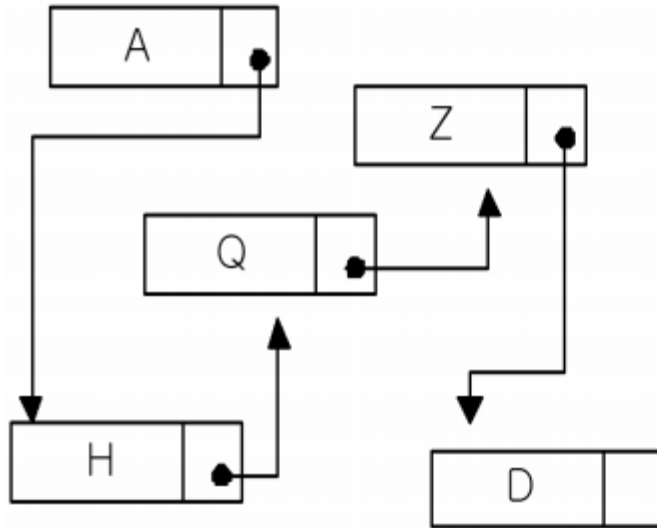
- 오버플로우의 처리 방법

- 선형 방법
- 이중 해싱 방법
- 동거자 연결 방법

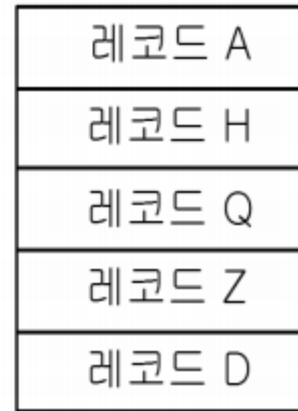
1.2.3 파일 구조에 따른 분류

● 링(Ring) 파일

➤ 레코드 사이의 관계를 포인터로서 연결시켜 주는 파일의 구성 방법



(a) 물리적 구성 형태



(b) 논리적 파일 형태

체인으로 된 레코드 구성

✓ 링 파일의 구조

- 단순한 한 방향 체인
- 꼬리 링크를 갖는 한 방향 체인
- 양방향 체인
- 양방향 링
- 헤드 레코드를 갖는 링
- 다중 링
- 코널 링

1.2.3 파일 구조에 따른 분류

- 순차 파일(Sequential File)
 - 색인 순차 파일(Indexed Sequential File)
 - 랜덤 파일(Random File)
 - 링 파일(Ring File)
-
- SAM file(Sequential Access Method file)
 - ISAM file(Indexed Sequential Access Method file)
 - DAM file(Direct Access file)

1.2.4 파일의 연산

- 일괄처리에서의 파일 연산

- 파일의 생성(creation)
- 파일의 복사(copy)
- 파일의 정렬(sort)
- 파일의 합병(merge)
- 파일의 대조(matching)
- 파일의 분배(distribution)
- 파일의 분배(distribution)
- 파일의 검색(search)
- 파일의 갱신(updating)

2장을 마치며.....

질의 & 응답

