

9.다중 키 파일

❖ 다중 키 화일의 개념

◆ 하나의 데이터 화일 - 다수의 접근 경로(키)

- (예) 학생 레코드 구조

학번	이름	학과	주민 등록 번호	입학 년도	지도 교수	주소
----	----	----	----------	-------	-------	----

◆ 구현 방법

- 데이터 중복 : 각 응용에 맞는 화일을 구성
 - ◆ 기억 공간 증가
 - ◆ 무결성(integrity) 유지 곤란
- 한 화일에 대한 다수의 접근 경로 구축
 - ◆ 역화일(inverted file) : 인덱스
 - ◆ 다중리스트 화일(multilist file) : 레코드 사이의 다중리스트

❖ 역 화일 구조

◆ 역화일 구조

- 인덱스를 이용하는 구조
- 역(inversion)
 - ◆ 인덱스와 데이터 레코드 화일을 연결

◆ 역 인덱스

- 데이터 화일에 있는 키 필드 값을 인덱스 키로 모두 포함
- 인덱스 엔트리 = (키 값, 레코드 포인터)
- 화일은 키 필드에 대해 전도(도치)되어 있음

▶ 학생 데이터 화일

레코드 주소	학번	이름	학과	주민 등록 번호	입학 년도	지도교수	주소
1	111111	이정진	컴퓨터	331004-*****	85		
2	112131	유근택	기계	490604-*****	86		
3	198121	이상구	전자	450816-*****	87		
4	201431	김성기	컴퓨터	241212-*****	86		
5	208432	용환승	기계	220321-*****	85		
6	291821	문봉기	자원	170108-*****	85		
7	300123	나민영	화공	300428-*****	86		
8	310103	이상돈	컴퓨터	340118-*****	85		
9	324196	황수찬	토목	340130-*****	85		
10	335812	한재술	전자	520101-*****	85		
11	386132	이규철	항공	510828-*****	86		
12	387122	홍봉희	항공	431204-*****	85		
13	415631	송성현	전기	271112-*****	86		
14	486212	김철수	전자	220717-*****	86		
15	511211	김연주	기계	471201-*****	88		
16	534216	유광석	토목	230413-*****	88		
17	535218	황시영	토목	330642-*****	88		
18	641230	이인기	전자	351230-*****	86		
19	686116	이규재	컴퓨터	501228-*****	86		
20	696123	안정화	화공	310203-*****	88		

▶ 레코드 주소를 이용한 주민등록번호 역 인덱스

주민 등록 번호	레코드 주소	레코드 주소	학번	이름	학과	입학 년도	지도 교수	주소
170108-*****	6	1	111111	이정진	컴퓨터	85		
220321-*****	5	2	112131	유근택	기계	86		
220717-*****	14	3	198121	이상구	전자	87		
230413-*****	16	4	201431	김성기	컴퓨터	86		
241212-*****	4	5	208432	용환승	기계	85		
271112-*****	13	6	291821	문봉기	자원	85		
300428-*****	7	7	300123	나민영	화공	86		
310203-*****	20	8	310103	이상돈	컴퓨터	85		
330642-*****	17	9	324196	황수찬	토목	85		
331004-*****	1	10	335812	한재술	전자	85		
340118-*****	8	11	386132	이규철	항공	86		
340130-*****	9	12	387122	홍봉희	항공	85		
351230-*****	18	13	415631	송성헌	전기	86		
431204-*****	12	14	486212	김철수	전자	86		
450816-*****	3	15	511211	김연주	기계	88		
471201-*****	15	16	534216	유광석	토목	88		
490604-*****	2	17	535218	황시영	토목	88		
501228-*****	19	18	641230	이인기	전자	86		
510828-*****	11	19	686116	이규재	컴퓨터	86		
520101-*****	10	20	696123	안정화	화공	88		

데이터 화일에서 키 값 제거. 역 인덱스에만 위치

▶ 역 인덱스의 구성

- ◆ 정렬 / 비정렬
- ◆ 역 인덱스의 키 순서 \neq 레코드 순서
- ◆ 인덱스 구조 : 테이블, 트리
- ◆ 직접 화일, 인덱스 된 순차화일 위에 구성

▶ 역화일

- ◆ 역 인덱스가 만들어지는 수에 따라
 - 완전 역화일 (Completely inverted file)
 - ◆ 모든 필드에 대한 역 인덱스
 - ◆ 데이터 파일은 존재 안 함
 - ◆ 실제 사용 예는 거의 없음
 - 부분 역화일 (Partially inverted file)
 - ◆ 몇 개의 필드에 대해서만 역인덱스 구성

▶ 역 인덱스 구성 방법의 대안

◆ 역 인덱스의 키 값을 데이터 레코드에서 제외

"학번이 335812인 학생의 주민등록번호는 ?"

① 데이터 파일 : 학번 335812인 학생 레코드의 주소

② 역인덱스 :

- 인덱스 엔트리 탐색
- 역인덱스의 순차 탐색
- 대단히 비효율적임

◆ 대안

- 간접 주소 기법 이용
- 인덱스 엔트리 = (보조키, 기본키)
 - ◆ 주소 값 대신 기본키 값 사용

▶ 기본 키를 이용한 주민등록번호 역 인덱스 구조의 변형

주민 등록 번호	학 번
170108-*****	291821
220321-*****	208432
220717-*****	486212
230413-*****	534216
241212-*****	201431
271112-*****	415631
300428-*****	300123
310203-*****	696123
330642-*****	535218
331004-*****	111111
340118-*****	310103
340130-*****	324196
351230-*****	641230
431204-*****	387122
450816-*****	198121
471201-*****	511211
490604-*****	112131
501228-*****	686116
510828-*****	386132
520101-*****	335812

- 데이터 화일의 물리적 재구성 또는 재조직 시 역 인덱스 화일에 영향 없음
- 직접주소의 경우 데이터 화일의 구조가 변경되면 역화일도 역시 재 구성되어야 함.

▶ 비유일 보조키에 대한 역 인덱스

◆ 학과 이름에 대한 역 인덱스

학 과	학 번
기 계	112131, 208432, 511211
자 원	291821
전 기	415631
전 자	198121, 335812, 486212, 641230
컴퓨터	111111, 201431, 310103, 686116
토 목	324196, 534216, 535218
항 공	386132, 387122
화 공	300123, 696213

◆ 문제점

- 키값 엔트리들의 정렬해야 하는가?
- 역인덱스의 구성은?
- 주소 기법 : 직접주소? 혹은 간접주소?

▶ 가변수의 포인터를 위한 인덱스 구현 방법

① 가변 길이의 인덱스 엔트리

학 과	길이	학 번			
기 계	3	112131	208432	511211	
자 원	1	291821			
전 기	1	415631			
전 자	4	198121	335812	486212	641230
컴퓨터	4	111111	201431	310103	686116
토 목	2	324196	534216	535218	
항 공	2	386132	387122		
화 공	2	300123	696213		

▶ 가변수의 포인터를 위한 인덱스 구현 방법

② 고정길이 엔트리 - 최대수의 엔트리 수용공간

학 과	학 번1	학 번2	학 번3	학 번4	학 번5
기 계	112131	208432	511211		
자 원	291821				
전 기	415631				
전 자	198121	335812	486212	641230	
컴퓨터	111111	201431	310103	686116	
토 목	324196	534216	535218		
항 공	386132	387122			
화 공	300123	696213			

▶ 가변수의 포인터를 위한 인덱스 구현 방법

③ 엔트리의 중복 - (보조키값, 하나의 주소) 쌍

학 과	학 번
기 계	112131
기 계	208432
기 계	511211
자 원	291821
전 기	415631
전 자	198121,
전 자	335812,
전 자	486212,
전 자	641230
컴퓨터	...
토 목	...
항 공	...
화 공	...

▶ 주소값 (기본키)의 정렬

◆ 오름차순으로 유지

- 신속한 레코드 검색
- 갱신 시의 추가 부담

◆ 인덱스만의 접근으로 응답 가능한 질의

- ① 주민등록번호가 410515-1036432인 학생이 있는가 ?
- ② 컴퓨터공학과에는 몇 명의 학생이 있는가 ?
- ③ 기계과에 속하는 학생의 학번들을 나열하라.
- 일반적인 키워드 검색이 이 경우에 속함
- ④ 주민등록번호가 241212-1032589인 학생의 학번은 무엇인가 ?

▶ 갱신

- ◆ 데이터 화일 + 역 인덱스
- ◆ 역 인덱스 화일 의 변화만을 보자
- ◆ 삽입

- 인덱스 엔트리 추가
 - ◆ (건축, 236514)추가
- 주소만 첨가
 - ◆ (컴퓨터, 623125)

◆ 삭제

- 주소만 제거
 - ◆ (컴퓨터, 111111) 제거
- 공백이면 엔트리 제거
 - ◆ (자원, 291821) 제거

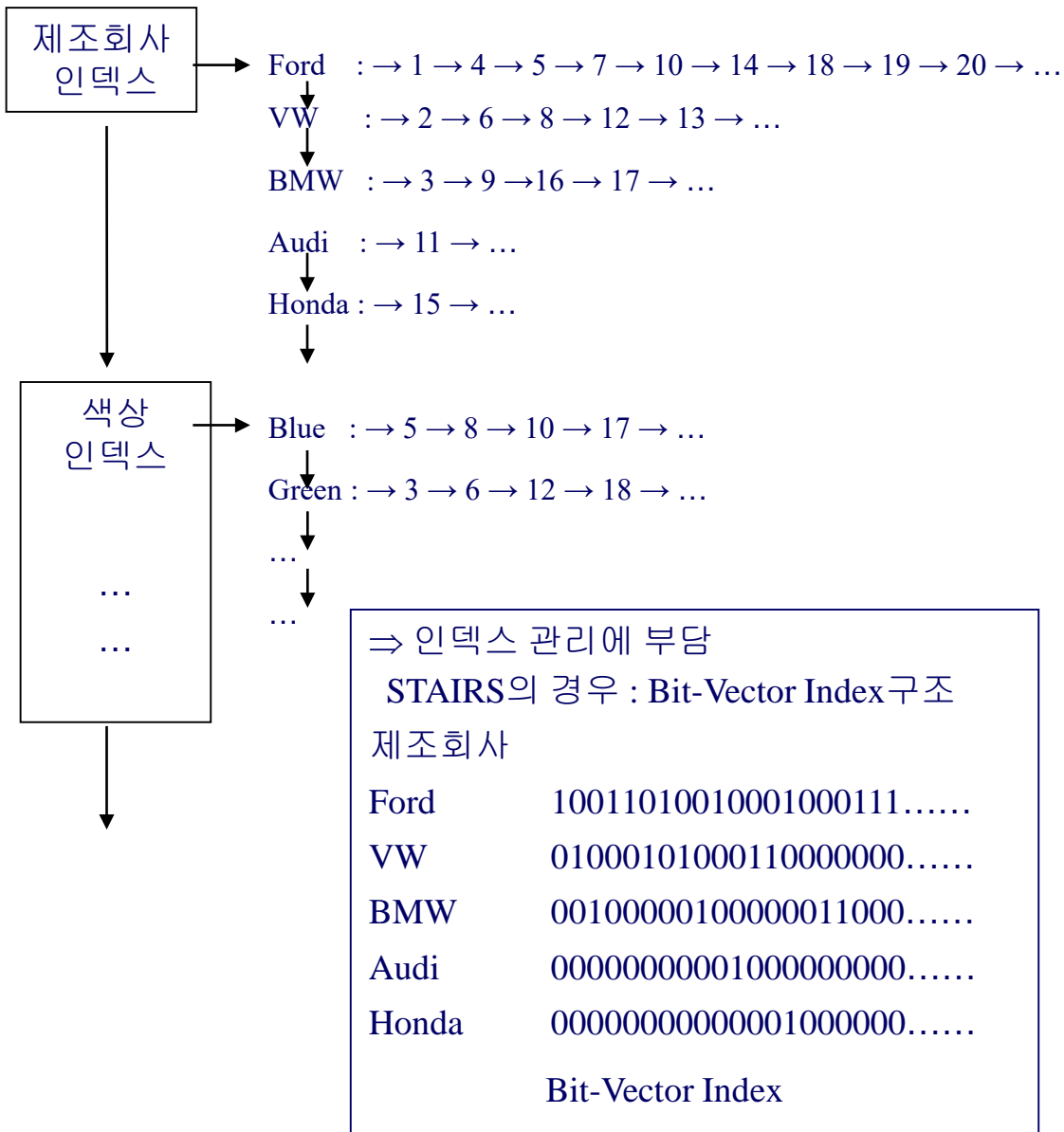
◆ 갱신

- 갱신 영향을 받는 모든 역 인덱스 화일에 삽입, 삭제 연산
 - ◆ (자원, 291821) 이 (기계, 291821) 로 변경되면?
 - (자원, 291821) 삭제, (기계, 291821) 추가

학 과	학 번
기 계	112131, 208432, 511211
자 원	291821
전 기	415631
전 자	198121, 335812, 486212, 641230
컴퓨터	111111, 201431, 310103, 686116
토 목	324196, 534216, 535218
항 공	386132, 387122
화 공	300123, 696213

▶ 역인덱스의 구현 사례

	제조회사	모델	색상	등록번호
1	Ford	Pinto	White	HORS4ME
2	VW	Bug	Red	SKIBNY
3	BMW	322i	Black	DADIOU1
4	Ford	Mustang	White	VALEGRL
5	Ford	Pinto	Blue	RATFACE
6	VW	Rabbit	Black	910VCD
7	Ford	Pinto	White	PACMAN
8	VW	Rabbit	Blue	BYE4NOW
9	BMW	320	Red	CMEGO
10	Ford	Mustang	Blue	DPGURU
11	Audi	5000	White	OU2LNCH
12	VW	Jetta	Black	1GWN821
13	VW	Bug	Green	BUG4AJS
14	Ford	Mustang	Red	1ABA800
15	Honda	Civic	Green	COMPSCI
16	BMW	320	White	4AUH20S
17	BMW	322i	Blue	번호
18	Ford	Tempo	Black	L8AGAIN
19	Ford	Pinto	Red	LUV2SKI
20	Ford	Mustang	Green	MYWHLS
...



❖ 다중 리스트 구조

- 역 인덱스
 - ◆ (키 값, 동일 키의 **모든** 레코드에 대한 포인터)
- 다중리스트 인덱스
 - ◆ (키 값, **하나의** 레코드에 대한 포인터)
 - ◆ 나머지 레코드
 - 데이터 파일에서 **연결리스트** 구조로 유지
 - ◆ 각 리스트의 헤드만 유지
 - 디렉토리 역할

다중 리스트 인덱스

학 과	포인터
컴퓨터	●
전 자	●

데이터 파일

이 름	학 과	
이 정 진	컴 퓨 터	↓
유 근 택	기 계 자	↓
이 상 구	전 자	↓
김 성 기	컴 퓨 터	↓
용 환 승	기 계 자	↓
문 봉 기	자 원 공	↓
나 민 영	화 공	↓
이 상 돈	컴 퓨 터	↓
황 수 찬	토 목 자	↓
한 재 술	전 공 공	↓
이 규 철	향 기 자	↓
이 홍 봉	성 현	↓
송 성 철	수 계	↓
김 연 주	전 목	↓
김 광 석	기 자	↓
유 광 시	토 공	↓
황 시 영	토 목	↓
이 인 기	전 자	↓
이 규 재	컴 퓨 터	↓
안 경 화	화 공	↓

▶ 다중리스트 인덱스와 화일의 관계

◆ 인덱스 구성 → 화일 구조 변경

- 인덱스 필드 수만큼 데이터 화일에 링크필드도 추가되어야 함
- 레코드의 포인터 필드 수 = 인덱스 수

◆ 고정길이 인덱스 엔트리

▶ 학과, 입학년도, 다중리스트에 대한 학생 데이터 화일

학 과		학 번
기 자 전 전 컴 토 항 화	계	112131
	원	291821
	기	415631
	자	198121
	터	111111
	목	324196
	공	386132
	공	300123

입학 년도	학 번
85	111111
86	112131
87	198121
88	511211

<입학 년도85>

레코드 주 소	학 번	이 름	학 과	다 음 학 번	주민 등록 번호	입학 년도	다 음 학 번
1	111111	이정진	컴퓨터	201431	331004-1032436	85	208432
2	112131	유근택	기 계	208432	490604-1533439	86	201431
3	198121	이상구	전 자	335812	450816-2987184	87	0
4	201431	김성기	컴퓨터	310103	241212-1032589	86	300123
5	208432	용환승	기 계	511211	220321-2033128	85	291821
6	291821	문봉기	자 원	0	170108-1036432	85	310103
7	300123	나민영	화 공	696123	300428-2267432	86	386132
8	310103	이상돈	컴퓨터	686116	340118-1097118	85	324196
9	324196	황수찬	토 목	534216	340130-2364432	85	335812
10	335812	한재술	전 자	486212	520101-1000611	85	387122
11	386132	이규철	항 공	387122	510828-2343125	86	415631
12	387122	홍봉희	항 공	0	431204-1334174	85	0
13	415631	송성현	전 기	0	271112-1315254	86	486212
14	486212	김철수	전 자	641230	220717-2036567	86	641230
15	511211	김연주	기 계	0	471201-1339451	88	534216
16	534216	유광석	토 목	535218	230413-1096418	88	535218
17	535218	황시영	토 목	0	330642-2684129	88	696123
18	641230	이인기	전 자	0	351230-2268914	86	686116
19	686116	이규재	컴퓨터	0	501228-1642517	86	0
20	696123	안경화	화 공	0	310203-2364968	88	0

▶ 데이터 레코드 구조

...	다중 리스트 키필드 1	다음 레코드 포인터 1	다중 리스트 키필드 2	다음 레코드 포인터 2	...
-----	-----------------	-----------------	-----------------	-----------------	-----

▶ 다중리스트 인덱스의 설계

◆ 고려 사항

- 인덱스 키값 엔트리의 정렬
- 인덱스의 구성방법
- 주소법 : 직접/간접
- 리스트의 데이터 레코드의 정렬 여부

▶ 레코드의 처리 과정

◆ 질의 유형

① 컴퓨터공학과에는 몇 명의 학생이 있는가 ?

- 레코드 수의 요구
- 인덱스 엔트리 (키 값, 포인터, **리스트 길이**)

학 과		학 번	길 이
기 자 전 전 컴 토 향 화	계 원 기 자 터 목 공 공	112131	3
		291821	1
		415631	1
		198121	4
		111111	4
		324196	3
		386132	2
		300123	2

입학 년도	학 번	길 이
85	111111	7
86	112131	8
87	198121	1
88	511211	4

▶ 레코드의 처리 과정

◆ 질의 유형

② 컴퓨터공학과에 속한 학생의 학번을 모두 검색하라.

- 다중리스트화일의 장점

③ 학번이 686116인 학생의 학과가 컴퓨터 공학과인가 ?

- 역화일구조는 데이터 접근 없이 가능

◆ 입학연도='01' and 학과='컴퓨터'인 학생의 이름은?

- 순차탐색 : 20 레코드

- 입학연도 리스트 탐색 : 7 레코드

- 학과 리스트 탐색 : 4 레코드

∴

i) 학과=컴퓨터인 레코드 접근

ii) 입학년도 필드 검사

▶ 갱신연산

- ◆ 데이터 파일 + 인덱스의 리스트 변경
- ◆ 삽입
 - 데이터 파일에 삽입
 - 인덱스 참조 → 리스트에 연결
 - 인덱스에 없으면 새 엔트리 삽입
- ◆ 삭제
 - 리스트에서 하나의 노드 삭제
 - 리스트의 유일 멤버 → 엔트리 삭제
- ◆ 갱신
 - 레코드 갱신
 - 인덱스 갱신
- ★ 리스트의 변경 ← 리스트의 구현 방법
 - 단순연결 리스트
 - 이중 연결 리스트
 - 단순 원형 리스트
 - 이중 원형 리스트

❖ 역 화일과 다중 리스트 화일의 비교

- 보조키에 대한 인덱스 유지

	역 인덱스	다중 리스트 인덱스
인덱스엔트리	동일 보조키의 모든 레코드	첫 레코드, 연결리스트
엔트리 길이	가변 길이	고정 길이
정렬	오버헤드	오버헤드
데이터 화일 구조	영향 없음	포인터 필드 첨가
차이점	<ul style="list-style-type: none"> • 질의 처리 능력 우월 (역 인덱스만의 접근으로 응답) • 프로그래머에 투명 : <ul style="list-style-type: none"> - 프로그래머는 인덱스의 존재를 몰라도 됨 	<ul style="list-style-type: none"> • 인덱스 관리 용이 (고정길이 엔트리) • 투명성 : 데이터 관리자의 별도작업 (레코드 끝에 포인터 별도저장) • 논리적 관계에 따른 접근 경로 제공 (상이한 타입의 레코드 간) • 한 화일에 다수의 정렬 순서 제공

