

4. 순차 파일

- 수록 순차 파일(Entry-Sequenced File, Pile)
- 키 순차 화일 (key-sequenced file)

데이터의 형태

SNUMBER = 1234 #SNAME = 홍길동 #SEX = 남 #IQ = 130;

SNUMBER=1234 #WEIGHT=60;

CITY = 서울 #POPULATION = 800만;

SNAME = 김철수 #HEIGHT = 170 #AGE = 30;

DEPARTMENT = 전산과 #NUMBER_OF_PROCESSOR = 10;

학번	이름	나이	본적	성
1243	홍길동	10	서울	남
1257	김철수	20	경기	남
1332	박영희	19	충청	여
1334	이기수	21	전라	남
1367	정미영	20	경상	여
1440	최미숙	21	강원	여

- 수록 순차 파일(Entry-Sequenced File, Pile)
- 키 순차 화일 (key-sequenced file)

❖ 순차 화일의 정의와 응용

◆ 수록 순차 파일(Entry-Sequenced File, Pile)

- 분석, 분류, 표준화과정을 거치지 않음
- 필드의 순서, 길이 등에 제한이 없음
- 레코드는 (이름-값)의 쌍으로 구성
- 병원의 의료 데이터 등에 적합
- 응용프로그램의 접근이 까다로움

SNUMBER = 1234 #SNAME = 홍길동 #SEX = 남 #IQ = 130;

SNUMBER=1234 #WEIGHT=60;

CITY = 서울 #POPULATION = 800만;

SNAME = 김철수 #HEIGHT = 170 #AGE = 30;

DEPARTMENT = 전산과 #NUMBER_OF_PROCESSOR = 10;

▶ 키 순차 화일 (key-sequenced file)

◆ 순차화일의 특징

- 저장 장치 내의 레코드 순서
= 레코드 리스트의 순서
- 레코드의 데이터 필드는 순서 집합 (ordered set)
 - 동일 순서의 데이터 필드값
 - 상이한 데이터 레코드로 구성 가능
- 화일 기술자(file descriptor) : 화일 정의 포함
- 오름차순(ascending) 혹은 내림차순(descending)
- 기본 키(primary key)
- 일괄처리(batch)용 응용
- 레코드 접근 방법 = 파일 내의 레코드 순서
 - 빠른 접근 (차위 레코드)
- 키 순차 화일의 구성 예

학번	이름	나이	본적	성
1243	홍길동	10	서울	남
1257	김철수	20	경기	남
1332	박영희	19	충청	여
1334	이기수	21	전라	남
1367	정미영	20	경상	여
1440	최미숙	21	강원	여

두 가지 이상의 상이한 레코드로 구성 가능

- ◆ 다른 정렬 키 → 다른 응용의 지원
동일한 내용의 다른 화일
- ◆ 상이한 형태의 레코드들로 구성 가능
 - 공통된 기능상의 목적
 - 논리적으로 밀접하게 관련
 - 가변 길이/고정 길이 레코드(공백을 채움)
- ◆ 예
 - 인사·급여 레코드

인사

레코드 형식	사원 번호	성 명	전화 번호	주 소			결혼 여부	성 별	우편 번호	교 육	
										기간	학위
1											

(a)

급여

레코드 형식	사원 번호	구 좌		기본급	배우자	부양 가족	장애자	수 당	재형 저축액	
		코드	번호							
2										

(b)

▶ 질의 적중 비율 (inquiry hit ratio)

질의에 응답위해 접근해야 할 레코드 수

=

파일 전체의 레코드 수

높은 적중 비율 → 순차 구조에 적합

낮은 적중 비율 → 직접파일 구조에 적합

❖ 순차 화일의 설계

◆ 설계 시 고려 사항

- 키 필드
 - ◆ 응용 요건에 따라 선정
- 레코드 내의 필드 배치
 - ◆ 활동적(active) / 비활동적 화일
 - 필드 타입 또는 레코드 어커런스에 따라

<마스터 인사 화일 형식>

사원 번호	이름	부서	직종 코드	프로젝트명	입사 일자	호봉	가입 보험	

<보조 인사 파일 형식>

사원 번호	최종 학위	출신교	졸업 년월	경력 코드	본적	호주 성명	

❖ 순차 화일의 설계 (중복제거)

학번	이름	성별	과목	성적
1	Lee	서울	M	a A
1	Lee	서울	M	b B+
1	Lee	서울	M	c A+
2	Kim	부산	F	a B
2	Kim	부산	F	b C+
2	Kim	부산	F	c B+

분해

학번	이름	성별
1	Lee	서울 M
2	Kim	부산 F

학번	과목	성적
1	a	A
1	b	B+
1	c	A+
2	a	B
2	b	C+
2	c	B+

Decomposition의 장점

- ① 데이터의 중복을 없앤다.
☆ 분해 전 : 36 cells \Rightarrow 분해 후 : $8+18=26$ cells
 - ② 데이터의 유지 관리가 용이
☆ Lee의 주소를 바꾼다고 가정
분해 전 : 3 tuples \Rightarrow 분해 후 : 1 tuple
 - ③ 무결성, 일관성 유지
☆ Lee의 주소를 바꾼다고 가정
 \Rightarrow 어느 한 tuple만 수정하면 불일치
- ☆ 수강을 하지 않는 학생은 삽입 불가능
 \Rightarrow 무결성에 위배

❖ 순차 화일의 생성

- ◆ 순서대로 저장 매체에 레코드를 수록
 - 데이터 수집
 - 컴퓨터가 읽을 수 있는 형태로 변환
 - 편집 : 수정(correction), 확인(verification)
 - ◆ pp.116~117
 - ◆ 무결성 제약조건
 - 개체무결성
 - 참조무결성
 - 도메인무결성
 - 편집된 데이터를 정렬
 - 기록
- ◆ 갱신 알고리즘을 이용한 방법

주민번호의 구조



◆ 오류검증번호(Check Digit)

- 고유번호 부여 시 고유번호의 타당성검증을 빠르고 쉽게 하기 위하여 부가하는 일종의 코드
- 주민번호, 군번, 계좌번호 등 대량의 데이터 처리 시 잘못된 번호 입력을 원천적으로 막아 시스템의 효율을 높이고자 함.

◆ 주민번호의 예

- 991201-179621?
 - ◆ 먼저 12개의 숫자에 각각 2,3,4,5,6,7,8,9,2,3,4,5를 차례로 곱한 후 나온 숫자를 모두 더함
 - $9 \times 2 + 9 \times 3 + 1 \times 4 + 2 \times 5 + 0 \times 6 + 1 \times 7 + 1 \times 8 + 7 \times 9 + 9 \times 2 + 7 \times 3 + 2 \times 4 + 1 \times 5 = 185$
 - ◆ 11로 나눈 나머지를 11에서 빼줌
 - $185 \bmod 11 = 9, 11 - 9 = 2$
 - ◆ 991201-1796212

❖ 순차 화일의 갱신

◆ 마스터 화일의 갱신 빈도

- 데이터 변화율 (data change rate)
- 마스터 화일의 크기
- 최신 데이터를 유지해야 할 필요성
- 화일 활동 비율 (file activity ratio)

$$= \frac{\text{트랜잭션에 영향을 받는 레코드 수}}{\text{마스터 화일의 총 레코드 수}}$$

◆ 갱신의 처리

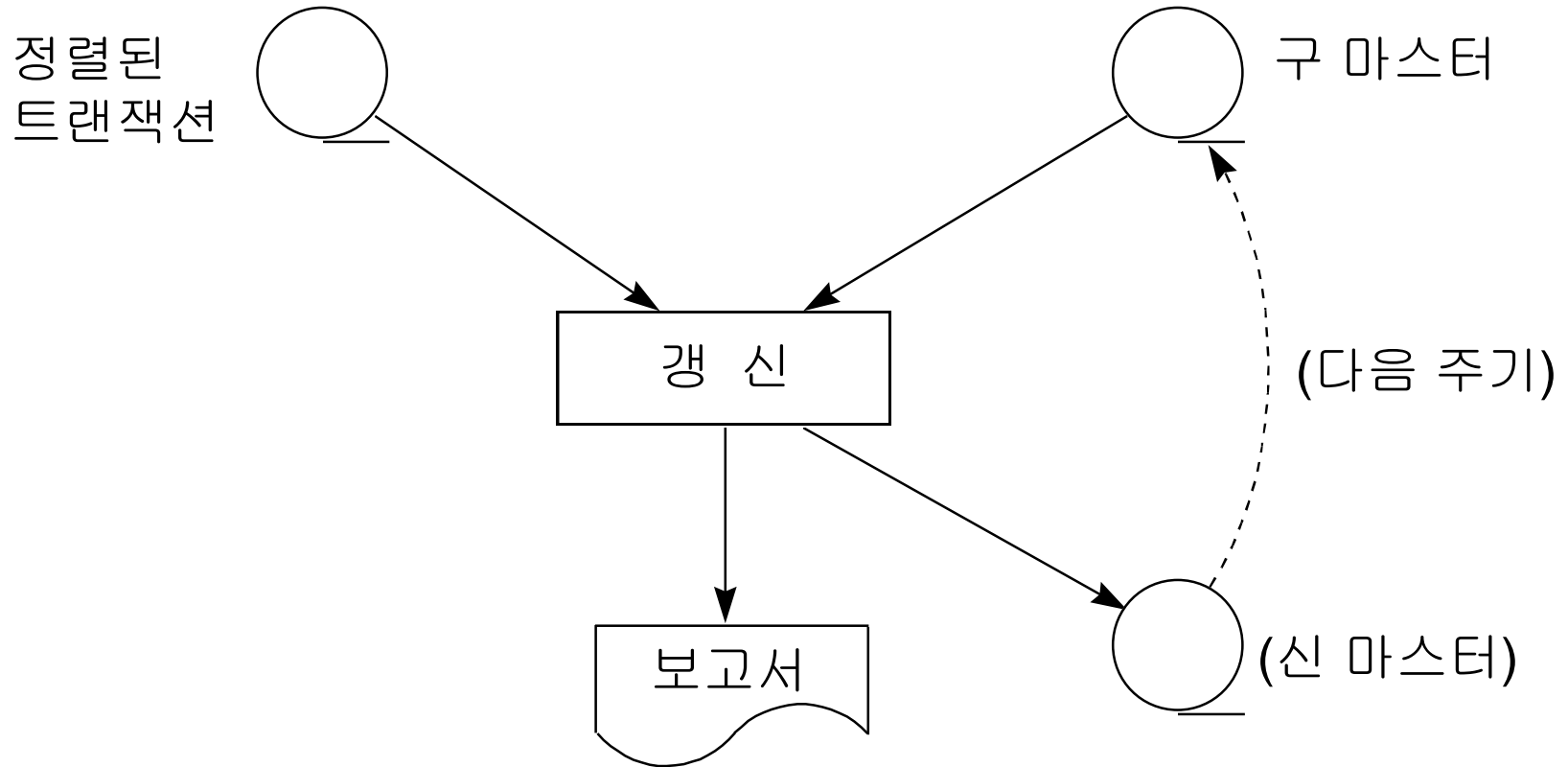
i) 트랜잭션 로그 화일

- accumulated (insertion) 화일

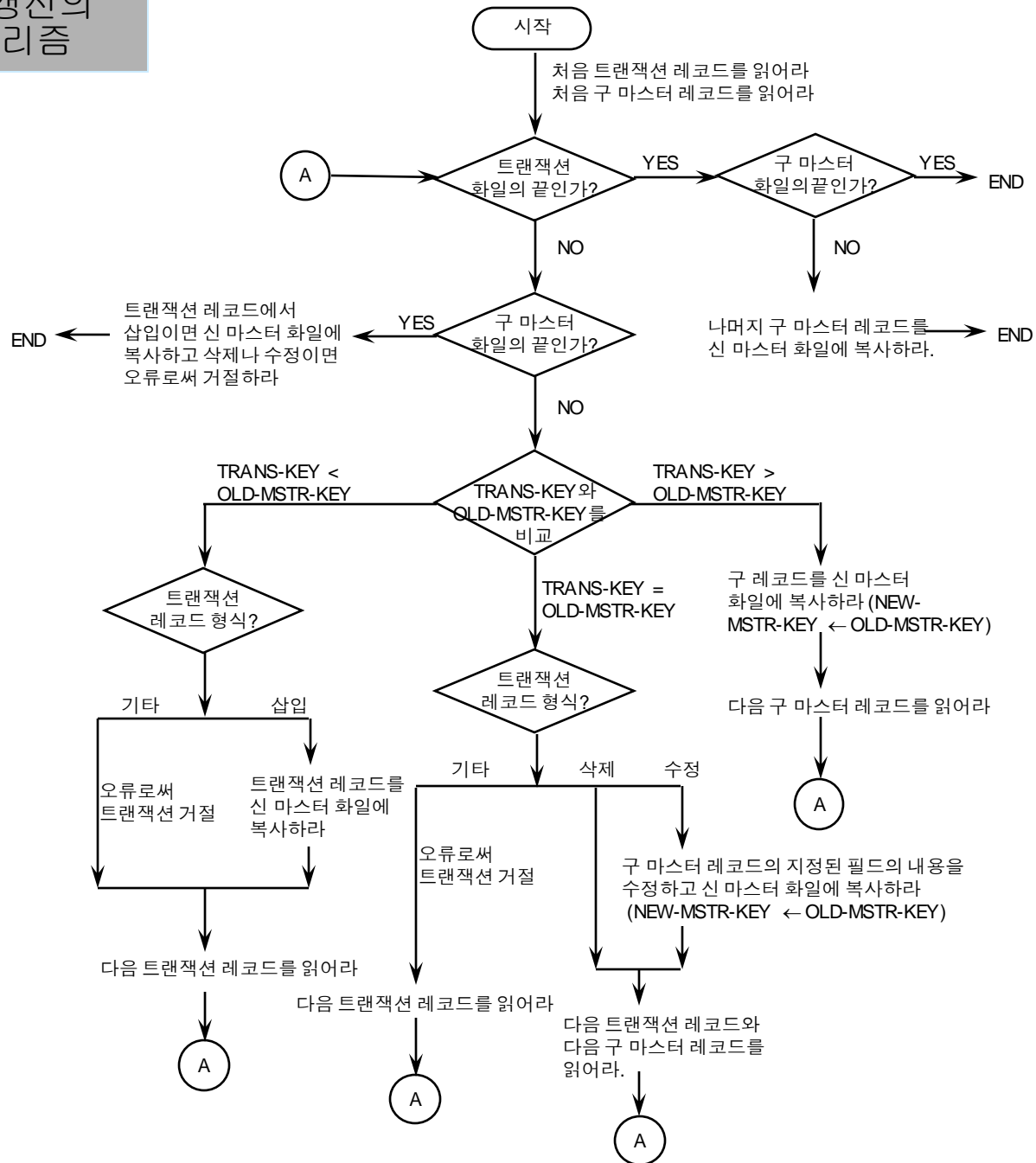
ii) 일괄 갱신

- ① 트랜잭션 로그 화일을 정렬 (키값에 따라)
- ② 새 순차 화일에 합병

▶ 순차화일 갱신의 시스템 흐름도



순차화일 갱신의 원시알고리즘



합병을 이용한 순차화일의 갱신 예제

구마스터화일



1	빨강
3	노랑
5	파랑
7	검정
9	초록1
(eof)	

트랜잭션화일



2	주황
3	
5	보라
(eof)	

신마스터화일



1	빨강
2	주황
5	보라
7	검정
9	초록

➤ 알고리즘

❑ 구마스터 키와 트랜잭션 키 비교

- ① = : 수정 혹은 삭제
- ② < : 구마스터 레코드를 신마스터에 복사
- ③ > : 트랜잭션 레코드를 신마스터에 복사

▶ 갱신 방법

◆ 트랜잭션 레코드의 구성

- i) 삽입 : <I, 새 키값, 나머지 필드값>
- ii) 삭제 : <D, 키 값>
- iii) 수정 : <C, 키 값, 필드의 새 값들>

<갱신을 위한 트랜잭션 레코드의 예>

트랜잭션 코드	사원 번호	성 명	주 소	부 서	전화 번호
I	12751	김 철수	당산 1동 128	경리과	

◆ 순차 화일의 갱신

TRANS-KEY : OLD-MSTER-KEY

= : 연산 실행 (수정, 삭제)

> : 구 마스터 레코드를 신 마스터 화일로 복사

< : 트랜잭션 레코드를 신 마스터 화일에 삽입

순차 화일 갱신 알고리즘

1. 수정 (단일 트랜잭션)

- 마스터 파일의 레코드에 대한 수정이 하나의 트랜잭션 레코드만으로 구성되는 경우.
- 트랜잭션 예

Old Master

품번	품명	재고량
1001	지우개	100
1005	연필	150
1008	볼펜	130

Transaction

품번	변동량
1005	-50
1006	+50



New Master

품번	품명	재고량
1001	지우개	100
1005	연필	100
1008	볼펜	130

[알고리즘 1] 마스터 화일 갱신 알고리즘 (단일 트랜잭션 변경)

```
Update_Master1(void) { /* 단일 트랜잭션 변경 */
    Get_Next_Trans();
    Get_Next_Master();
    while ( !(master_key == sentinel && trans_key == sentinel)) {
        if (master_key < trans_key) { /* 트랜잭션 없음 */
            output_mater_record_to_new_master();
            Get_Next_Master();
        }
        else
            if (master_key == trans_key) {
                make_change_in_master_record();
                output_master_record_to_new_master();
                Get_Next_Master();
                Get_Next_Trans();
            }
        else { /* master_key > trans_key */

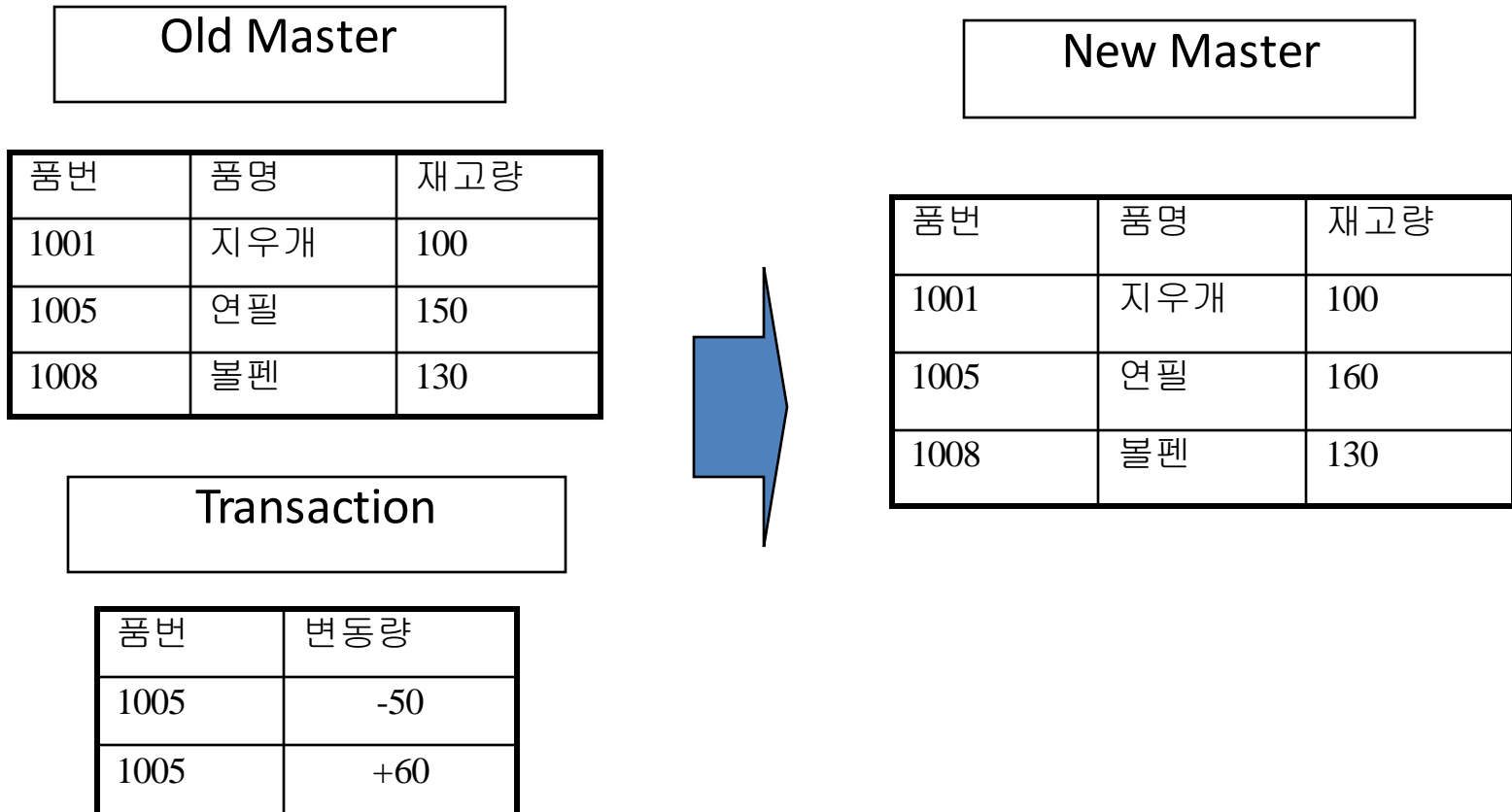
            printf("No Matching master record for trans key.\n");
            Get_Next_Trans();
        }
    }

    Get_Next_Master(void) {
        if(( c = getchar() ) == EOF)
            master_key = sentinel_value;
        else
            input_master_record;
    }

    Get_Next_Trans(void) {
        if (( c = getchar() ) == EOF )
            trans_key = sentinel_value;
        else
            input_transaction_record;
    }
}
```

2. 수정 (다중 트랜잭션)

- 마스터 파일의 레코드에 대한 갱신이 여러 개의 트랜잭션 으로 구성되는 경우.
- 트랜잭션 예



[알고리즘 2] 마스터 화일 갱신 알고리즘 (다중 트랜잭션 변경)

```
Update_Master2(void)          /* 다중 트랜잭션 변경 */
{
1  Get_Next_Trans();
2  Get_Next_Master();
3  while( !(master_key == sentinel && trans_key == sentinel))
4      if(master_key < trans_key)          /* 트랜잭션 없음 */
5      {
6          output_mater_record_to_new_master();
7          Get_Next_Master();
8      }
9      else
10         if(master_key == trans_key)
11         {
12             make_change_in_master_record();
13             Get_Next_Trans();
14         }
15         else          /* master_key > trans_key */
16         {
17             printf("No Matching master record for trans key.\n");
18             Get_Next_Trans();
19         }
20 }
```

단일 트랜잭션 make_change_in_master_record(); output_master_record_to_new_master(); Get_Next_Master(); Get_Next_Trans();
--

3. 추가 + 수정 (다중 트랜잭션)

- 마스터 화일의 레코드에 대한 갱신이 다음과 같을 때
 - ① 각 트랜잭션 키에 대해 한번 또는 그 이상의 변경
 - ② 각 트랜잭션 키에 대해 한번 추가하고 한번 또는 그 이상 변경
 - ③ 각 트랜잭션 키에 대해 여러 번 추가하고 한번 또는 그 이상 변경
- 갱신코드를 갖는 새로운 필드 정의
 - 코드 A : 마스터 화일에 추가하는 트랜잭션
 - 코드 C : 현존하는 마스터 레코드에 변경하는 트랜잭션

Old Master

품번	품명	재고량
1001	지우개	100
1005	연필	150
1008	볼펜	130

Transaction

갱신코드	품번	품명	변동량
A	1001	지우개	+50
C	1005		-50
C	1005		+20
A	1006	삼각자	+50
C	1006		-10

New Master

품번	품명	재고량
1001	지우개	100
1005	연필	120
1006	삼각자	40
1008	볼펜	130

오류

[알고리즘 3] 마스터 화일 갱신 알고리즘 (다중 트랜잭션 추가 포함)

```
Update_Master3(void)          /* 다중 트랜잭션 삽입 변경 */
{
    Get_Next_Trans();
    Get_Next_Master();
    while( ! (master_key == sentinel && trans_key == sentinel))
        if (master_key < trans_key)          /* 트랜잭션 없음 */
        {
            output_mater_record_to_new_master();
            Get_Next_Master();
        }
        else
            if (master_key == trans_key)
            switch(update_code)
            {
                case 'A' :      printf("Duplicate add\n");
                                Get_Next_Trans();
                                break;
                case 'C' :      make_change_in_master_record();
                                Get_Next_Trans();
                                break;
                default :        printf("Invalid update code\n");
                                Get_Next_Trans();
            }
            else                /* master_key > trans_key */
                Make_New_Record(); /* 새 레코드를 생성 */
    }
}
```


```

Make_New_Record( )      /*새로운 레코드를 생성함 */
{
    switch(update_code)
    {
        case 'A' :    build_new_record_from_trans_record();
                      new_key = trans_key;
                      Get_Next_Trans();
                      while(trans_key != sentinel && trans_key == new_key)
                      {
                          switch(update_code)
                          {
                              case 'A' :    printf("Duplicate add\n");
                                              break;
                              case 'C' :    make_change_in_new_record();
                                              break;
                              default :    printf("Invalid update code\n");
                                              break;
                          }
                          Get_Next_Trans();
                      }
                      output_new_record_to_new_master();
                      break;

        case 'C' :    printf("No matching master record for trans key\n");
                      Get_Next_Trans();
                      break;

        default :    printf("Invalid update code\n");
                     Get_Next_Trans();
                     break;
    }
}

```



4. 추가 + 수정 + 삭제 (다중 트랜잭션)

- 마스터 화일의 레코드에 대한 갱신이 다음과 같을 때
 - ① 각 트랜잭션 키에 대해 한번 삭제 수행
 - ② 각 트랜잭션 키에 대해 여러 번 삭제 수행
 - ③ 각 트랜잭션 키에 대해 추가 한번과 삭제 한번 수행
 - ④ 각 트랜잭션 키에 대해 한번 또는 그 이상의 변경과 삭제 한번 수행
 - ⑤ 각 트랜잭션 키에 대해 추가 한번, 한번 또는 그 이상의 변경, 삭제 한번 수행
- 갱신 코드
 - 추가-A
 - 변경-C
 - 삭제-D
- 트랜잭션 예

갱신코드	id번호	제조회사	차형	모델	총운행거리	색깔
D	F1					
A	GM7	Pontiac	2 DR	Fiero	1,250	orange
C	GM7				5,000	
D	GM7					
A	GM7	Pontiac	2 DR	Fiero	1,500	navy
D	H3					
C	T1				7,800	
D	T1					

[알고리즘 4] 마스터 화일 갱신 알고리즘4 (다중 트랜잭션 Full)

```
Update_Master4(void)          /* 다중 트랜잭션 삽입 삭제 변경 */
{
    Get_Next_Trans();
    Get_Next_Master();

    while( ! (master_key == sentinel && trans_key == sentinel))
        if(master_key < trans_key)          /* 트랜잭션 없음 */
        {
            output_mater_record_to_new_master();
            Get_Next_Master();
        }
        else
            if(master_key == trans_key)
                switch(update_code)
                {
                    case 'A' :      printf("Duplicate add");
                                    Get_Next_Trans();
                                    break;
                    case 'C' :      make_change_in_master_record();
                                    Get_Next_Trans();
                                    break;
                    case 'D' :      Get_Next_Master();
                                    Get_Next_Trans();
                                    break;
                    default :        printf("Invalid update code");
                                    Get_Next_Trans();
                }
            else                    /* master_key > trans_key */
                Make_New_Record();
}
```

```

Make_New_Record()
{
    switch(update_code)
    {
        case 'A' : build_new_record_from_trans_record();
                   new_key = trans_key;
                   delete_record = FALSE;
                   Get_Next_Trans();
                   while(trans_key != sentinel && trans_key == new_key && ! delete_record)
                   {
                       switch(update_code)
                       {
                           case 'A' : printf("Duplicate add");
                                       break;
                           case 'C' : make_change_in_new_record();
                                       break;
                           case 'D' : delete_record = TRUE;
                                       break;
                           default : printf("Invalid update code");
                       }
                       Get_Next_Trans();
                   }
                   if( ! delete_record)
                       output_new_record_to_new_master();
                   break;

        case 'C' :
        case 'D' : printf("No matching master record for trans key");
                   Get_Next_Trans();
                   break;

        default : printf("Invalid update code");
                  Get_Next_Trans();
                  break;
    }
}

```

4번째 트랜잭션
D GM7의 경우

5 사례 연구

■ 렌트카 영업소 예

- 가정 : 렌트카 마스터 화일이 비었다
- 마스터 화일에 추가할 데이터를 갖는 트랜잭션 화일 T1

<Ç¥ 5.1> ·»Æ®Ä« ¿μ¾÷¼ÒÀÇ Æ®·£Àè¼Ç È-Äï T1

°»½ÄÄÚμá	id'øÈ£	Á!Á¶È»ç	Â÷Çü	„ðμ”	ÃÑ¿iÇà°Á,®	»ö±ò
A	C1	Dodge	2 DR	Omni	25,000	grey
A	C2	Dodge	2 DR	Aapen	7,000	tan
A	F1	Ford	2 DR HB	Escort	54,000	white
A	F2	Lincoln	4 DR	Continental	38,000	black
A	F3	Ford	2 DR	Thunderbird	35,000	blue
A	GM1	Cadillac	4 DR	Fleetwood	9,000	red
A	GM2	Oldsmobile	4 DR	Delta 88	28,050	blue
A	GM3	Chevrolet	2 DR	Camaro	33,000	silver
A	GM4	Cadillac	2 DR	Cimarron	63,000	maroon
A	GM5	Oldsmobile	4 DR	98	11,000	green
A	H1	Honda	4 DR	Accord	32,000	yellow
A	H2	Honda	2 DR HB	Accord	11,250	brown
A	T1	Toyota	2 DR HB	Celica	3,400	white

<Ç¥ 5.2> ·»Æ®Ä« çµ¼÷¼ÒÀÇ ,¶½°ÁÍ È-ÀĬ

id'øĖĖ	Á:Á¶Ė,»ç	Â÷Çü	,ðµ"	ÃÑ¿îÇà°Á,®	»ö±ò
C1	Dodge	2 DR	Omni	25,000	grey
C2	Dodge	2 DR	Aapen	7,000	tan
F1	Ford	2 DR HB	Escort	54,000	white
F2	Lincoln	4 DR	Continental	38,000	black
F3	Ford	2 DR	Thunderbird	35,000	blue
GM1	Cadillac	4 DR	Fleetwood	9,000	red
GM2	Oldsmobile	4 DR	Delta 88	28,050	blue
GM3	Chevrolet	2 DR	Camaro	33,000	silver
GM4	Cadillac	2 DR	Cimarron	63,000	maroon
GM5	Oldsmobile	4 DR	98	11,000	green
H1	Honda	4 DR	Accord	32,000	yellow
H2	Honda	2 DR HB	Accord	11,250	brown
T1	Toyota	2 DR HB	Celica	3,400	white

<Ç¥ 5.3> Æ®·£Àè¼Ç È-ÀĬ T2

°»½ÁÄÜµí	id'øĖĖ	Á:Á¶Ė,»ç	Â÷Çü	,ðµ"	ÃÑ¿îÇà°Á,®	»ö±ò
A	GM6	Chevrolet	4 DR	Cimarron	11,250	brown

<Ç¥ 5.4> Æ®·£Àè¼Ç È-ÀĬ T2·Î°»½ÁÈÄÀÇ ,¶½°ÁÍ È-ÀĬ

id'øĖĖ	Á:Á¶Ė,»ç	Â÷Çü	,ðµ"	ÃÑ¿îÇà°Á,®	»ö±ò
C1	Dodge	2 DR	Omni	25,000	grey
C2	Dodge	2 DR	Aapen	7,000	tan
F1	Ford	2 DR HB	Escort	54,000	white
F2	Lincoln	4 DR	Continental	38,000	black
F3	Ford	2 DR	Thunderbird	35,000	blue
GM1	Cadillac	4 DR	Fleetwood	9,000	red
GM2	Oldsmobile	4 DR	Delta 88	28,050	blue
GM3	Chevrolet	2 DR	Camaro	33,000	silver
GM4	Cadillac	2 DR	Cimarron	63,000	maroon
GM5	Oldsmobile	4 DR	98	11,000	green
GM6	Chevrolet	4 DR	Cimarron	11,250	brown
H1	Honda	4 DR	Accord	32,000	yellow
H2	Honda	2 DR HB	Accord	11,250	brown
T1	Toyota	2 DR HB	Celica	3,400	white

<Ç¥ 5.4> Æ®·£Àè¼Ç È-ÄÏT2·Î»½ÄÈÄÀÇ, ¶½°ÁÍ È-ÄÏ

id¹ø££	Á:Á¶£,»ç	Â:Çü	,ðµ"	ÃÑ¿îÇà°Á,®	»ö±ò
C1	Dodge	2 DR	Omni	25,000	grey
C2	Dodge	2 DR	Aapen	7,000	tan
F1	Ford	2 DR HB	Escort	54,000	white
F2	Lincoln	4 DR	Continental	38,000	black
F3	Ford	2 DR	Thunderbird	35,000	blue
GM1	Cadillac	4 DR	Fleetwood	9,000	red
GM2	Oldsmobile	4 DR	Delta 88	28,050	blue
GM3	Chevrolet	2 DR	Camaro	33,000	silver
GM4	Cadillac	2 DR	Cimarron	63,000	maroon
GM5	Oldsmobile	4 DR	98	11,000	green
GM6	Chevrolet	4 DR	Cimarron	11,250	brown
H1	Honda	4 DR	Accord	32,000	yellow
H2	Honda	2 DR HB	Accord	11,250	brown
T1	Toyota	2 DR HB	Celica	3,400	white

<Ç¥ 5.5> Æ®·£Àè¼Ç È-ÄÏT3

°»½ÄÄÜµà	id¹ø££	Á:Á¶£,»ç	Â:Çü	,ðµ"	ÃÑ¿îÇà°Á,®	»ö±ò
D	F1					
A	GM7	Pontiac	2 DR	Fiero	1,250	orange
C	GM7				5,000	
D	GM7					
A	GM7	Pontiac	2 DR	Fiero	1,500	navy
D	H3					
C	T1				7,800	
D	T1					

<표 5.6> 트랜잭션 화일 T3로 갱신후의 마스터 화일

id번호	제조회사	차형	모델	총운행거리	색깔
C1	Dodge	2 DR	Omni	25,000	grey
C2	Dodge	2 DR	Aapen	7,000	tan
F2	Lincoln	4 DR	Continental	38,000	black
F3	Ford	2 DR	Thunderbird	35,000	blue
GM1	Cadillac	4 DR	Fleetwood	9,000	red
GM2	Oldsmobile	4 DR	Delta 88	28,050	blue
GM3	Chevrolet	2 DR	Camaro	33,000	silver
GM4	Cadillac	2 DR	Cimarron	63,000	maroon
GM5	Oldsmobile	4 DR	98	11,000	green
GM6	Chevrolet	4 DR	Cimarron	11,250	brown
GM7	Pontiac	2 DR	Fiero	1,500	navy
H1	Honda	4 DR	Accord	32,000	yellow
H2	Honda	2 DR HB	Accord	11,250	brown

❖ 순차 화일의 저장 장소

◆ 저장 매체

- 순차 접근 저장 매체 : 카드, 테이프
- 직접 접근 저장 매체 : 디스크
 - ◆ 익스텐트(Extent)
 - 하나의 파일이 차지하는 연속된 디스크 구역
 - 하나의 실린더에 다 채워질 때까지 저장
 - 남은 다음 실린더로 이동하여 다 채워질 때까지 저장
 - 만일 한면의 트랙을 채우고 다음 면의 트랙을 채워나가면?
 - 하나의 트랙이 채워질 때 마다 암이 이동을 해야함

◆ 직접 접근 장치가 바람직한 경우

- 순차 접근 장치의 부족
- 디스크에 기록되어야 하는 화일
 - ◆ 스푼링(Simultaneous Peripheral Operation Online)
 - 주변장치와 컴퓨터 내부장치 간에 데이터를 전송할 때 처리 지연을 단축하기 위해 디스크를 완충 기억장치로서 사용하는 것.

