Introduction to ***File processing***

강의 개요 및 수업진행방향

담당교수: 이수철 (dakterlee@gmail.com)

- COVID-19에 따른 수업방식 변경안내

  1. 본 과목은 대면 방식으로 진행해야 되나 부득이 하게 오프라인으로 진행하게 되었습니다.

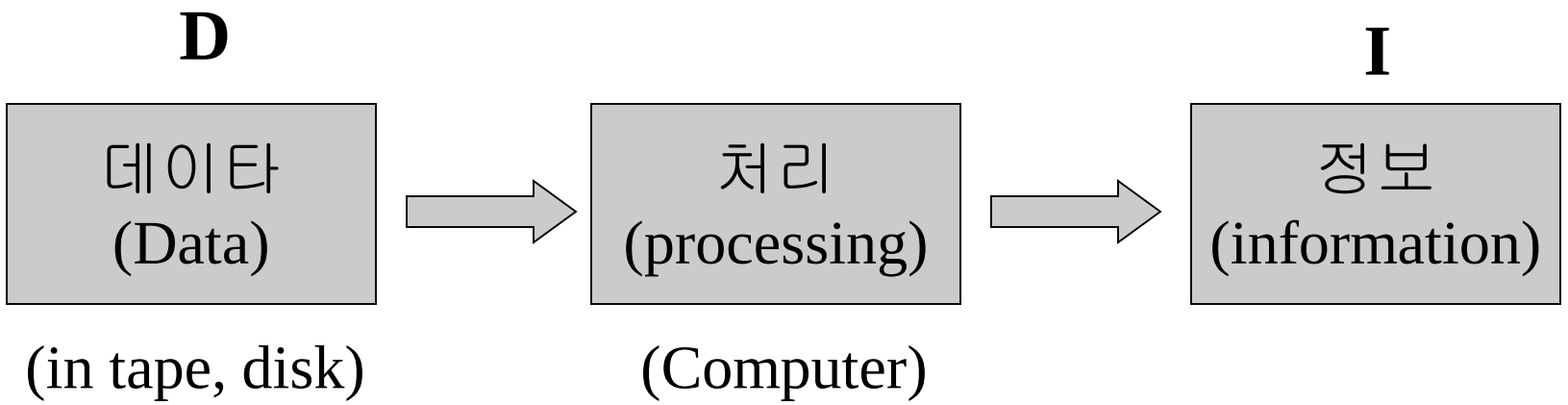
  2. 오프라인으로 단순히 ppt+음성을 입히려고 하였으나, 크게 효과적이지 못할 것으로 판단하여 이번주는 강의자료를 읽고 report를 제출하는 방식으로 진행하려 합니다.

  3. 제1장의 강의자료와 책을 보신 후 제1장의 연습문제를 풀어 제출하시기 바랍니다. (31일 11시 59분 까지)

1. 화일의 기본개념

❖ 화일의 종류

- 정보 ≠ 데이타



I = P(D)

▶ 중요 용어 (1)

- **데이타 필드 (field), 애트리뷰트 (attribute), 데이타 항목 (item)**
  - 이름을 가진 논리적 데이타의 최소 단위
  - 특정 객체(object, entity)의 한 성질의 값
- **레코드 타입 (record type)**
  - 한 종류의 객체를 기술하는, 논리적으로 서로 연관된 데이타 필드(항목)들의 구조
  - 엔티티 타입
- **레코드 어커런스(record occurrence)**
  - 한 레코드 타입의 인스턴스(instance)
  - 레코드 타입의 각 필드에 따라 실제 값이 들어가 어떤 특정 사물을 나타내는 것

▶ 중요 용어 (2)

- **화일(file)**
  - 보조기억장치에 저장된 같은 종류의 (논리적으로 연관) 레코드 집합
    - disk(RAM), tape(SAM)
  - 공통 응용 목적을 위해 함께 저장된 데이타
    - 예 : 급여 계산, 인사 기록, 재고 관리, 과학 기술 계산

- **데이타의 집합을 왜 화일로 구성하는가?**
  ① 주기억장치에 전부 적재하기에 너무 많은 양
  ② 프로그램은 특정시간에 데이타의 일부만 접근
    - 데이타 전부를 주기억장치에 한꺼번에 저장시킬 필요가 없음
  ③ 데이타를 특정 프로그램의 수행과 독립적으로 보관
    - 데이타의 독립성(independency) 유지

▶ 화일의 분류 (1)

➔ 기능에 따라

- ◆ 마스터 화일 (master file)
- ◆ 트랜잭션 화일 (transaction file)
- ◆ 보고서 화일 (report file)
- ◆ 작업 화일 (work file)
- ◆ 프로그램 화일(program file)

(1) 마스터 화일 (master file)

- **어느 한 시점에서 조직체의 업무에 관한 정적인 면을 나타내는 데이타의 집합**
  - 예(제조 회사) : 급여 마스터 화일, 고객 마스터 화일, 인사 마스터 화일, 재고 마스터 화일, 자재 요청 마스터 화일
- **비교적 영구적(permanent)인 데이타 또는 역사적 데이타 (historical status data)를 포함**

★ **사전 화일 (dictionary file)**
- 마스터 화일의 특수한 종류
- 데이타의 기술(description) - 타입, 크기, 이름, 사용

(2) 트랜잭션 화일 (transaction file)

- 마스터 화일의 변경 내용을 저장
- 마스터 화일에 새 레코드 추가, 현존 레코드를 제거 또는 수정하기 위한 데이타

★ **트랜잭션 (transaction)**

- 논리적인 작업 단위
- 하나의 건수로 처리하는 작업

트랜잭션의 예 (1)

처리 순서는 중요하지 않지만
두 개의 UPDATE 문이 모두 정상적으로
실행되어야 함

그림 10-1 **트랜잭션의 예1 : 계좌이체 트랜잭션**

트랜잭션의 예 (2)

INSERT 문과 UPDATE 문이 모두 정상적으로 실행되어야 상품주문 트랜잭션이 성공적으로 수행됨

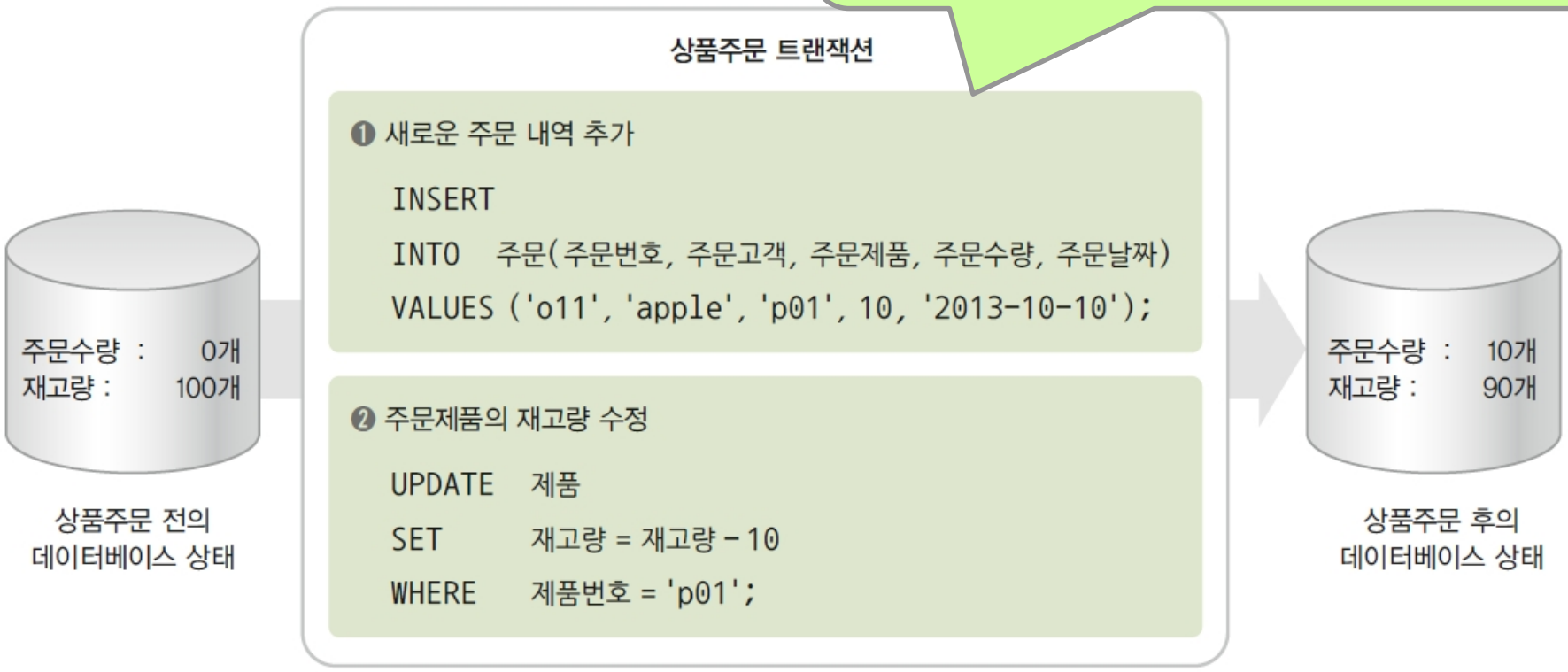
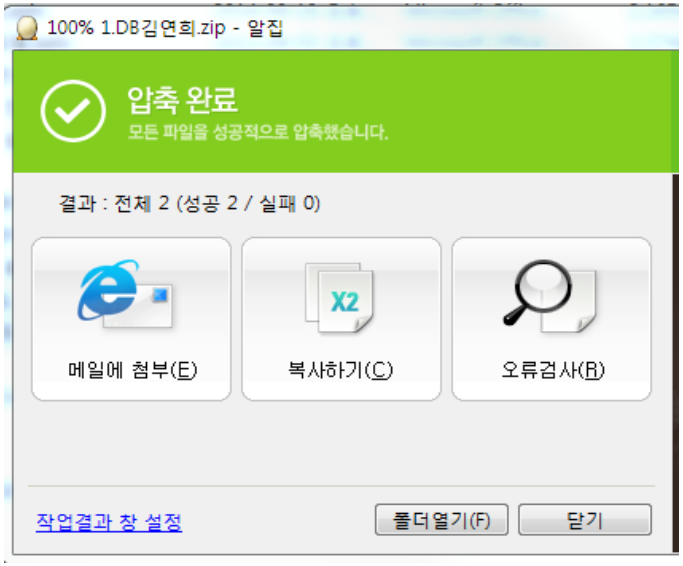
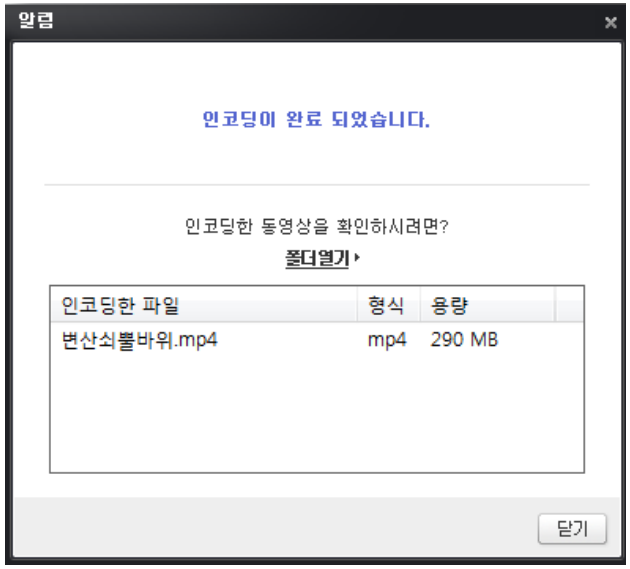


그림 10-2 **트랜잭션의 예2 : 상품주문 트랜잭션**

(3) 보고서 화일 (report file)

- 사용자에게 보고서로 보이기 위해 일정한 형식을 갖춘(formatted) 데이타를 저장
  - 하드카피(hard copy), 단말 장치 화면

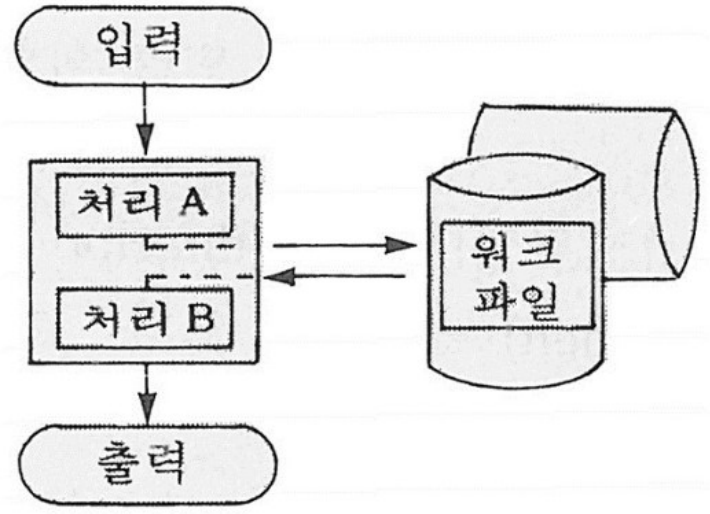




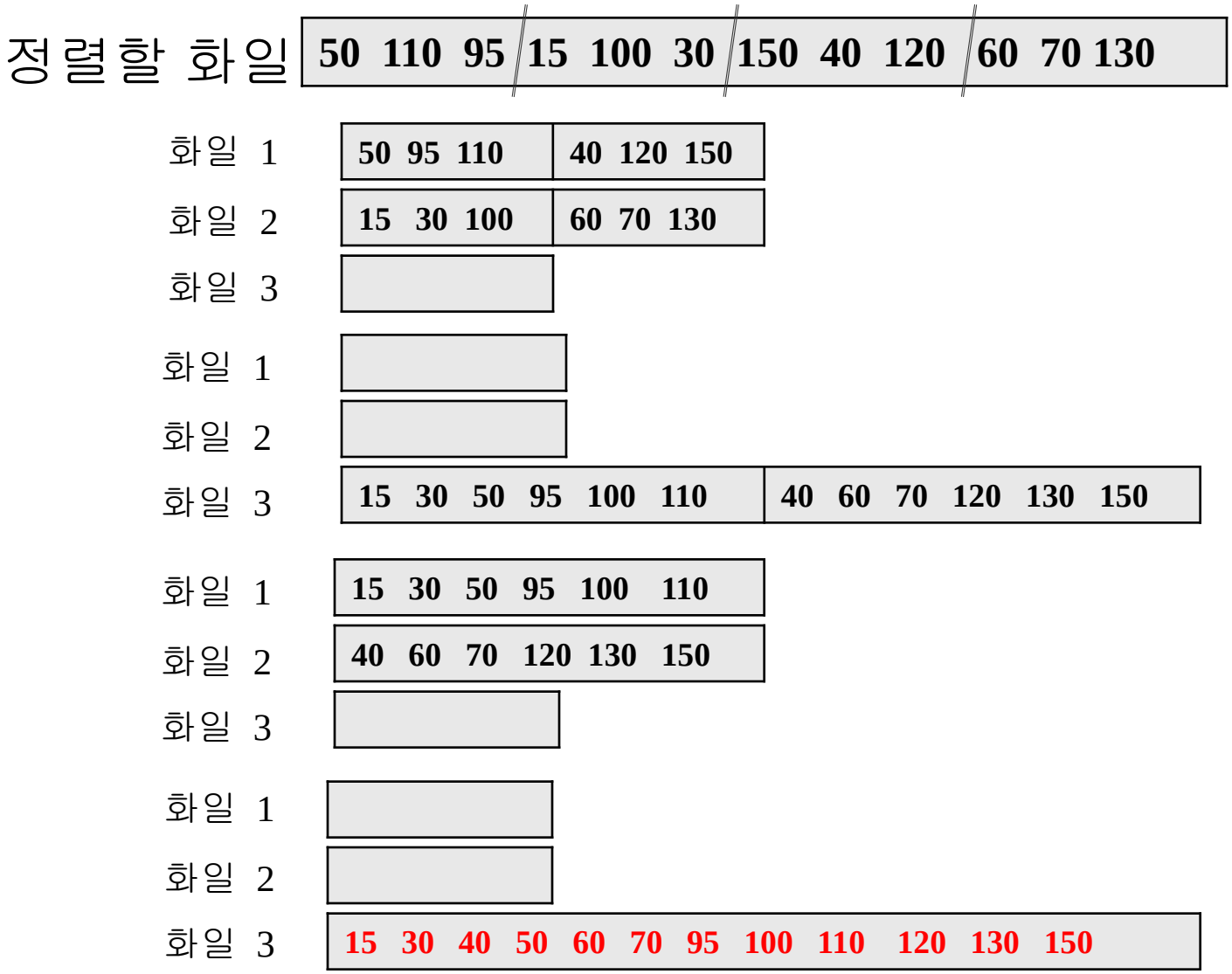
(4) 작업 화일 (work file)

◆ 시스템에 있는 임시 화일(temporary file)

프로그램 실행중에 어떤 처리 결과를 일단 보조 기억 장치로 출력하고 다음 처리 단계에서 그 출력을 입력으로서 처리를 행하는 경우가 있다.
이와 같이 처리 단계에 작업(work)용으로 사용하는 보조 기억 장치 상의 파일을 작업용 파일 또는 작업 파일이라고 한다.
분류(sorting) 등의 프로그램에 있어서도 이러한 작업용 파일이 사용된다.
[네이버 지식백과] 작업 파일 [work file] (컴퓨터인터넷IT용어대사전, 2011. 1. 20., 일진사)



▶ 4 개의 런에 대한 2-원 합병 예제



(5) 프로그램 화일 (program file)

- 데이타를 처리하는 명령어들을 포함
- 고급언어(**COBOL, PASCAL**), 어셈블리어, 기계어, 작업 제어 언어(**job control language**) 등으로 작성

▶ 화일의 분류 (2)

➔ 프로그램의 화일 접근 목적에 따라

(1) 입력 화일 (input file)

- 프로그램이 읽기만 함

(2) 출력 화일 (output file)

- 프로그램이 기록만 하기 위해 사용
- 프로그램에 의해 작성

(3) 입/출력 화일 (input/output file)

- 프로그램의 실행 중 읽기도 하고 기록하기도 함

▶ 화일 조직의 기본 개념 (1)

- **키 (key) :**
  - 레코드를 식별하는데 사용되는 레코드 필드
    - 기본키(primary key) : 데이타 레코드를 유일하게 식별하고 저장하는 기억장소를 결정하는데 사용되는 레코드 필드
    - 보조키 (secondary key) : 나머지 레코드 필드 중에서 레코드를 접근하는데 사용되는 레코드 필드

키의 개념

고객 릴레이션

고객아이디	고객이름	나이	등급	직업	적립금
apple	김현준	20	gold	학생	1000
banana	정소화	25	vip	간호사	2500
carrot	원유선	28	gold	교사	4500
orange	정지영	22	silver	학생	0

고객 릴레이션의 기본키

주문 릴레이션

주문번호	주문고객	주문제품	수량	단가	주문일자
1001	apple	진짜우동	10	2000	2013-01-01
1002	carrot	맛있는파이	5	500	2013-01-10
1003	banana	그대로만두	11	4500	2013-01-11

주문 릴레이션의 기본키

주문 릴레이션의 외래키

외래키 속성과 그것이 참조하는 기본키 속성의 이름은 달라도 되지만 도메인은 같아야 한다.

그림 5-10 외래키의 예 : 고객 릴레이션과 주문 릴레이션

키의 개념

그림 5-11 **학생 상담 데이터베이스 스키마**

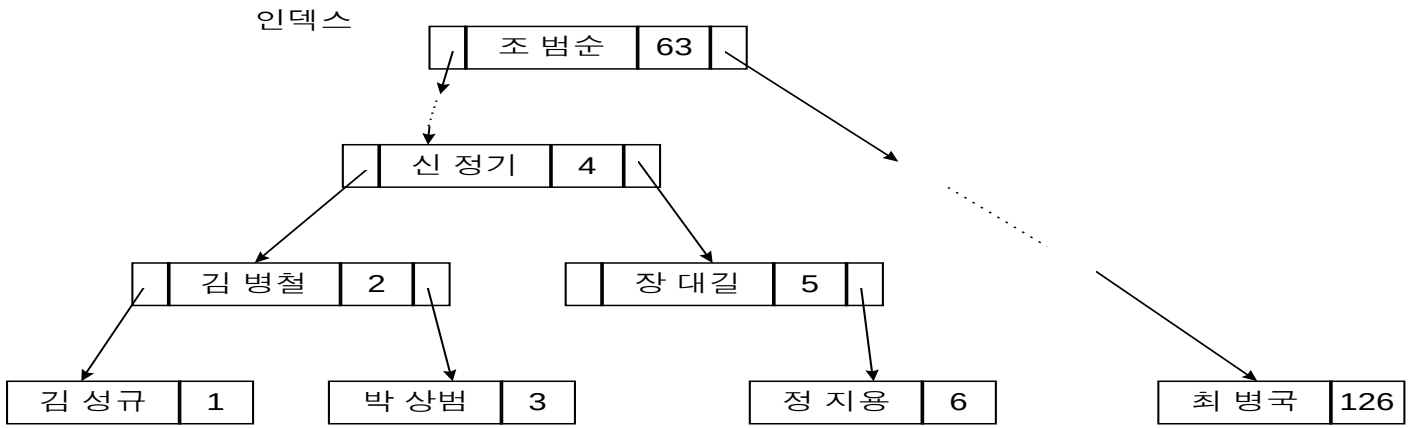
하나의 릴레이션에는 외래키가 여러 개 존재할 수도 있고
외래키를 기본키로 사용할 수도 있다.

▶ 화일 조직의 기본 개념 (2)

- **인덱스 (index) :**
  - 화일의 특정 필드에 대한 접근 효율을 높이기 위해서 만드는 보조적인 구조
    - 기본 인덱스 (primary index) : 기본키를 포함한 필드들에 대한 인덱스
    - 보조 인덱스(secondary index) : 기본 인덱스 이외의 인덱스
    - 집중 인덱스 (clustered index) : 데이타 레코드의 순서가 인덱스의 엔트리 순서와 동일하거나 유사하도록 유지하는 인덱스
    - 비집중 인덱스(unclustered index) : 집중 형태가 아닌 인덱스
    - 밀집인덱스 (dense index) : 데이타 레코드 하나에 대해 적어도 하나의 인덱스 엔트리를 구성해놓은 인덱스 = 역인덱스(inverted index)
    - 희소 인덱스 (sparse index) : 레코드 그룹 또는 데이타 블록별로 하나씩 인덱스를 만들어 두는 인덱스

▶ 인덱스된 순차 화일의 예

- 이원 탐색 트리의 인덱스 구조를 가진 인덱스된 순차 화일의 예



순차 데이타 화일

1	김 성규		
2	김 병철		
3	박 상범		
4	신 정기		
5	장 대길		
6	정 지용		
⋮	⋮		
63	조 범순		
⋮	⋮		
126	최 병국		

❖화일의 조직

- **화일의 데이타 레코드를 표현, 저장하는 기법**

(1) 순차화일 (sequential file)

- 데이타를 저장 장치의 물리적 순서대로 저장
- 각 레코드 내의 데이타 항목들은 모두 동일한 순서로 존재

(2) 인덱스된 순차 화일 (indexed sequential file)

- 데이타에 대한 인덱스(임의 접근)와 순차 접근 제공
- 인덱스, 순차 데이타 구역, 오버플로우 구역 (overflow area)으로 구성

(3) 직접 화일(direct file)

- 레코드의 키값이 연산 루틴에 의해 그 키값을 갖는 레코드의 주소로 변환

❖화일의 조직 (계속)

(4) 다중 키 화일 (multi-key file)

- 인덱스를 통해서만 데이타 접근
- 탐색 매개 변수가 되는 데이타 항목 : key

(5) 다차원 화일 (multidimensional file)

- 인덱스의 탐색키가 여러개의 필드를 포함하는 복합키(composite key)에 대한 인덱스를 지원하는 파일
- 여러개의 필드가 모여 하나의 키 역할을 수행

▶ 화일 조직 기법의 특성

1) 화일의 레코드 순서를 결정

- 보조기억장치 내에서 레코드의 물리적 순서
  - 정렬 순서 : 정렬 키 필드의 값
  - 임의 순서

2) 어떤 필드에 특정 값을 갖는 레코드를 탐색하는데 필요한 연산의 집합을 결정

- 저장 장치의 운영(operational) 특성
  → 화일의 조직에 영향

  - disk → 직접접근 저장장치
    (DASD : Direct Access Storage Device)
  - tape → 순차접근 저장장치

▶ 화일 사용의 형식

- **일괄처리(batch) 형식**
  - 마스터 화일 효율적으로 접근하도록 트랜잭션들을 구성함
  - 트랜잭션들을 그룹화하여 처리하는 성능이 주요 관심사

- **대화(interactive) 형식**
  - 트랜잭션이 터미널에 도착하는 대로 구성하고 처리함
  - 개개 트랜잭션의 처리 성능이 주요 관심사

▶화일에 대한 기본 연산

(1) 생성

(2) 기록(갱신, 삽입, 삭제)

(3) 판독(화일 이름, 블록 명세)

(4) 삭제

(5) 개방과 폐쇄(버퍼의 할당과 반환)

(1) 생성 (creation)

- 데이타 조직의 설계

  - skeleton design : data definition

- 데이타 수집(collection)과 확인(validation)

- 데이타 적재(loading)
  - 공간 할당 → 데이타가 한꺼번에 적재
  - 한 번에 한 레코드씩 구성

(2) 기록 (write)

- 마스터 화일의 내용을 기록

  i) 레코드 내용의 변경 (update)

  ii) 새로운 레코드의 삽입(insert)

  iii) 레코드 삭제(delete)

(3) 판독 (read)

- **마스터 화일의 내용을 판독**
  - 화일 이름, 판독해야 할 블록 명세
  - 디렉토리 조사(기록 연산과 비슷)

(4) 삭제 (delete)

- **화일의 삭제**
  - 화일 위치 검색
  - 디스크 공간 반환, 디렉토리 엔트리 삭제

(5) 개방과 폐쇄 (open, close)

- **화일의 개방**
  - 연산을 수행하기 위한 준비 단계
  - 판독, 기록 가능
  - 버퍼 할당

- **화일의 폐쇄**
  - 디스크에 버퍼 데이타 기록
  - 버퍼 반환
  - 화일에 대한 사용권한 반납

❖화일 구조 선정 요소

- **주기억 장치**
  - 비교 연산 등 주요연산의 수행 횟수로 평가
  - 데이타 접근시간은 모두 일정한 것으로 가정

- **보조 저장 장치**
  - 데이타 접근 시간이 메인 메모리에 비해 매우 길다
  - 보조 저장 장치의 접근 횟수가 프로그램 성능 평가 요소

    —> **화일 구조 선정의 중요성**

❖화일 구조 선정 요소

- 화일 구조 선정 요소

  (1) 가변성

  (2) 활동성

  (3) 사용빈도수

  (4) 응답 시간

  (5) 화일 크기

  (6) 화일 접근 유형

(1) 가변성(volatility)

- **화일의 성격**
  - 내용이 변하지 않는 정적 화일 (과거의 기록)
  - 내용이 자주 변하는 동적 화일 (현재의 상황 데이타)

- **가변성(volatility)**
  - 전체 레코드 수에 대해 추가되거나 삭제되는 레코드 수
  - 가변성이 높은 동적 화일은 빠른 접근과 갱신이 필요

(2) 활동성(activity)

- **화일의 활동성**
  - 주어진 기간 동안(대개 한번의 트랜잭션)에 화일의 총 레코드 수에 대해 접근한 레코드 수의 비율
  - 활동성이 높으면 순차 화일 구조 유리

(3) 사용 빈도수 (frequency of use)

- **화일의 사용 빈도수**
  - 일정 기간 동안의 화일의 사용 빈도수
  - 가변성과 활동성에 밀접히 관련

- **사용 빈도수와 화일 구조**
  - 제한된 화일 접근 방법이 사용 빈도수에 장애
  - 빈도수가 낮으면 순차 화일 구조 유리
  - 빈도수가 높으면 임의 접근 구조 유리

(4) 응답 시간(response time)

- **응답 시간과 화일 구조**
  - 검색이나 갱신에 대해 요구하는 지연 시간
  - 빠른 응답 시간 조건에는 임의 접근 방법 선택
  - 정렬된 키를 이용한 순차 접근 방법 가능

(5) 화일 크기(file size)

- **화일 크기와 화일 구조**
  - 레코드 수와 각 레코드 길이가 화일 크기 결정
  - 시간이 지남에 따라 화일 크기 성장

    (레코드 길이 확장, 레코드 수 증가)
  - 성장을 유연하게 수용할 수 있는 구조 필요
    - 정적해싱 vs. 동적해싱

(6) 화일 접근 유형

- **화일 접근 유형과 화일 구조**
  - 연산의 유형과 접근 형식에 따라 화일 구조 결정

ex) 1. 판독 위주 접근 ? 갱신 위주 접근 ?

2. 순차 접근 주도 ? 임의 접근 주도 ?