

1. 화일의 기본개념

가톨릭대학교
황병연 교수

❖ 학습 내용

- ◆ 파일 처리
- ◆ 파일의 종류
- ◆ 파일의 연산
- ◆ 파일 구조 선정 요소

❖ 파일 처리

- 데이터를 **파일**로 구성하고 관리하는 기술은 정보시스템의 가장 중요한 요소로서 컴퓨터 시스템이 활용되면서부터 연구 대상이 되었고, 급속히 발전해 옴
- 정보화 시대에서 **데이터 공용**은 **정확성**과 **보안** 측면에서 엄격한 관리가 요구되며, 파일의 구성 방법과 처리 기법에 영향을 줌

→ **처리 기술 중요**

- **ICBM ?**

Inter Continental Ballistic Missile: 대륙간탄도미사일

→

▶ 빅데이터

■ 빅데이터 ?

수십 TB(terabyte) 이상

실시간, 스트림 처리

정형(테이블), 반정형(XML), 비정형(SNS)

정확하고 신뢰할 수 있어야 함

문제 해결을 위한 의사 결정에
활용될 만한 가치

이해하기 쉽게 그림이나 도표로 시각화

가변성을 인식하고 수집과 분석 작업에서
데이터의 원래 의미가 그대로 반영될 수
있도록 노력해야 함

(Volume)

(Velocity)

(Variety)

(Veracity)

(Value)

(Visualization)

(Variability)

3V

5V

7V

▶ 정보와 데이터

▶ 정보(Information)와 데이터(Data)가 = or ≠ ?



현실 세계로부터 관찰이나 측정을 통해서
수집된 이나

수치, 스트링, 텍스트, 이미지, 그래픽스



의사결정(Decision-making)을 할 수 있게 하는
지식(Knowledge)으로
데이터의 이나
데이터

▶ 데이터와 정보 처리



의미 있는 정보가 되기
위한 2가지 조건은?

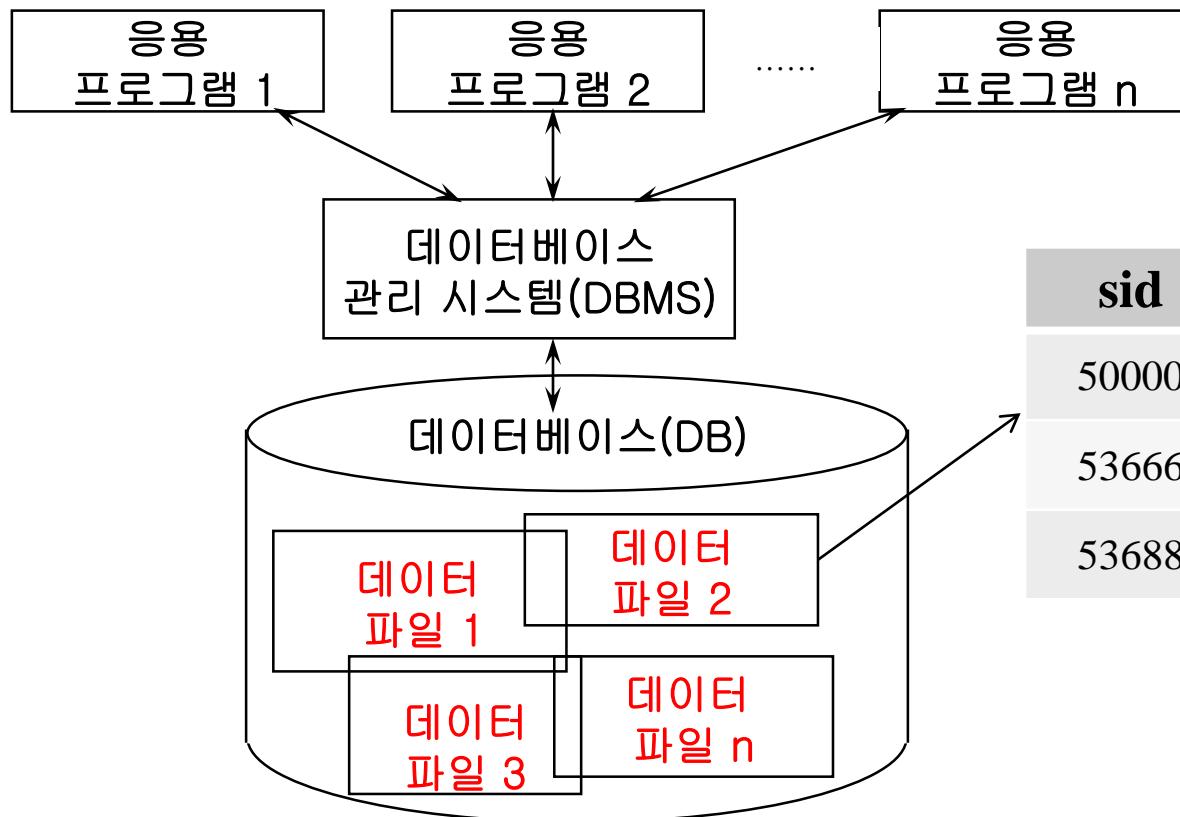


▶ 데이터 저장과 자료 처리

- ◆ 일반적으로 프로그램에 의해 처리될 대용량의 데이터는 어디에 저장되는가? Main Memory or Disk?
- ◆ 컴퓨터는 ()의 데이터를 ()에 가져와서 처리?
- ◆ 컴퓨터로 자료를 처리할 때 중요한 이슈는?
 - 데이터를 어떻게 효율적으로 ()하는가?
 - 데이터를 어떻게 효율적으로 ()하는가?

▶ 파일과 데이터베이스

- 파일은 하나의 파일 자체로도 중요하지만 고차원적인 데이터베이스 시스템을 구현하는 하부구조로도 중요함



sid	name	login
50000	Dave	dave@cs
53666	Jones	jones@cs
53688	Smith	smith@ee

STUDENT

▶ 화일의 종류

- ◆ 디스크에 저장하는 데이터는 크게 무엇으로 구분되는가?
⇒
- ◆ 하나의 파일(file) 이란?
 - 어떤 공통적인 응용 목적(급여, 인사, 재무 등)을 가지고 보조기억장치에 저장된 데이터 레코드(record) 집합
 - 레코드는 서로 연관된 데이터 필드(field)들로 구성
 - 필드는 이름을 가진 논리적 데이터의 최소 단위이며 애트리뷰트(attribute) 또는 데이터 항목(item)이라고도 함

- 데이터베이스 시스템을 구현하는 하부구조

<u>bid</u>	bname	color
101	Interlake	red
103	Titanic	blue
.		

Boats

▶ 화일의 종류

◆ 화일 구조란?

- 디스크에 저장할 데이터의 **표현(representation)**과 데이터를 접근하기 위한 **연산(operation)**의 조합
- 응용으로 하여금 데이터를 판독, 기록, 변경할 수 있게 함
- 어떤 탐색 조건에 맞는 데이터를 검색하거나 어떤 특정 순서로 데이터를 판독할 수 있게도 함
- 어떻게 파일 구조를 설계하느냐에 따라 시스템이 시간을 허비하지 않고 효율적으로 데이터를 처리할 수 있게 함
- 한 경우에 최적인 것이 다른 경우에는 최악이 될 수도 있기 때문에 다양한 타입의 데이터와 응용의 다양성이 파일 구조 설계를 어렵게 함

▶ 화일의 종류

- ◆ 데이터의 집합을 왜 디스크 화일로 구성하는가?
 - 주기억장치에 전부 적재하기에 데이터 양이 너무 많다.
 - 프로그램은 특정시간에 데이터 집합의 일부만 접근한다.
 - ⇒ 데이터 전부를 주기억장치에 한꺼번에 저장시킬 필요가 없음
 - 데이터를 특정 응용 프로그램의 실행과 별도로 저장시켜 데이터의 독립성(independency)을 유지한다.
 - ⇒ 여러 응용 프로그램이 ()하기 쉬움

▶ 파일의 분류

◆ 기능에 따라

- 마스터 파일 (master file)
- 트랜잭션 파일 (transaction file)
- 보고서 파일 (report file)
- 작업 파일 (work file)
- 프로그램 파일 (program file)
- 텍스트 파일 (text file)

◆ 프로그램의 파일 접근 목적에 따라

- 입력 파일 (input file)
- 출력 파일 (output file)
- 입출력 파일 (input/output file)

▶ 기능에 따른 파일의 분류

◆ 마스터 파일 (master file)

- 삽입, 삭제, 수정을 통해 영속적 데이터 레코드를 포함하고 있는 파일
- 보통 파일이라고 하면 마스터 파일을 의미
- 예(제조 회사) : 급여/고객/인사/재고/자재 요청 마스터 파일
- 마스터 파일의 내용은 **현재성**을 정확히 유지해야 함

▶ 기능에 따른 파일의 분류

◆ 트랜잭션 파일 (transaction file)

- 마스터 파일에 적용할 변경 내용을 모아 저장한 파일
- 예) 인사 마스터 파일을 변경해야 될 때, 인사 변동의 내용만을 담은 데이터 파일
- 마스터 파일에 새로운 레코드 삽입(insert), 현존 레코드의 삭제(delete), 현존 레코드 수정(update)을 담은 데이터 포함

◆ 트랜잭션 (transaction) 이란?

- 하나의 논리적인 작업 단위
- 하나의 건수로 처리되어야 하는 분리될 수 없는 단일 작업
예)

▶ 기능에 따른 파일의 분류

◆ 보고서 파일 (report file)

- 사용자에게 정보 검색의 결과를 보여주기 위해 일정한 형식을 갖춘(*formatted*) 데이터를 저장하고 있는 파일
- 하드카피(hard copy) 보고서 출력하거나 단말 장치 화면에 디스플레이

◆ 작업 파일 (work file)

- 어느 한 프로그램에서 생성된 출력 데이터를 다른 프로그램의 입력 데이터로 사용하기 위해 임시로 만들어 사용하는 파일 (temporary file)
- 시스템이 자동으로 만드는 작업 파일
⇒ 예) ()을 위한 파일

▶ 기능에 따른 파일의 분류

◆ 프로그램 파일 (program file)

- 데이터를 처리하기 위한 명령어들을 저장하고 있는 파일
- 고급언어(C, JAVA 등), 어셈블리어, 기계어와 같은 저급 언어
- 원시 코드(source code)와 컴파일화된 목적 코드(object code)

◆ 텍스트 파일 (text file)

- 문자 숫자(alphanumeric)와 그래픽 데이터를 포함하고 있는 파일
- 텍스트 편집기의 입력과 출력으로 사용

▶ 접근 목적에 따른 파일의 분류

◆ 입력 파일 (input file)

- 프로그램이 판독을 위해 접근하는 파일
- 세율 테이블 파일은 종합 소득에 대한 세금 계산용 입력 파일
- 원시 프로그램 파일은 컴파일러의 입력 파일

◆ 출력 파일 (output file)

- 프로그램이 기록을 위해 접근하는 파일
- 보고서 파일, 목적 코드 프로그램 파일

◆ 입-출력 파일 (input-output file)

- 프로그램의 실행 중 판독과 기록을 위해 접근하는 파일
- 급여 마스터 파일에 있는 급여 총액 항목은 세율 계산시 입력으로 사용되고, 연봉 계산 후 저장할 때 출력을 위해 접근

❖ 화일의 연산

◆ 화일 사용시 두 가지 중요한 면

- 화일 **사용**의 형식
- 화일 **연산**의 성격

▶ 파일 사용의 형식

◆ 일괄처리(batch) 형식

- 마스터 파일을 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

▶ 파일에 대한 기본 연산

◆ 파일 생성

- 데이터 정의, 적재

◆ 파일 기록

- 레코드 삽입, 삭제, 갱신

◆ 파일 판독

- 파일의 이름과 판독해야 할 블록을 명세함

◆ 파일 삭제

- 파일 제거

◆ 파일의 개방과 폐쇄

- 버퍼의 할당과 반환

▶ 생성 (creation)

- ◆ 데이터를 어떻게 조직할 것인가 골격에 대한 설계
 - 데이터 정의(data definition)
- ◆ 데이터 수집하고 저장 장치에 저장
 - 데이터 적재/loading)
- ◆ 파일 생성하려면
 - 파일을 위한 공간이 할당된 후
 - 새로운 파일에 대한 엔트리가 딕터리에 만들어짐
 - 딕터리 엔트리에는 파일 이름, 파일의 위치 등 정보 포함

▶ 기록 (write)

- ◆ 마스터 파일의 내용을 기록하기 위해서 파일 이름, 기록할 데이터를 명세해야 함
- ◆ 파일의 이름이 주어지면 시스템은 디렉터리를 조사해서 파일의 위치를 찾아냄
- ◆ 파일 기록 연산
 - 새로운 레코드의 삽입(insert)
 - 기존 레코드 삭제(delete)
 - 기존 레코드 내용의 변경 (update)

▶ 판독 (read)

- ◆ 마스터 파일의 내용을 판독하기 위해서 파일 이름과 판독해야 할 블록을 명세해야 함
- ◆ 파일의 이름이 주어지면 디렉터리를 조사해서 파일의 위치와 판독해야 할 레코드의 디스크 주소를 찾아냄

▶ 삭제 (delete)

◆ 파일의 삭제

- 디렉터리에서 명세된 파일 위치 검색
- 다른 파일이 사용할 수 있도록 디스크 공간 반환
- 디렉터리 엔트리 삭제

▶ 개방과 폐쇄 (open, close)

◆ 파일의 개방 (fopen)

- 연산을 수행하기 위한 준비 단계
- 파일의 시작점에서 판독, 기록 가능
- 메인 메모리에 버퍼 할당

◆ 파일의 폐쇄 (fclose)

- 디스크에 버퍼 데이터 기록
- 버퍼 반환

▶ 주기억 장치와 보조 저장 장치

◆ 주기억 장치

- 최대 비교 연산 횟수로 평가
- 데이터 접근시간은 모두 일정한 것으로 가정

◆ 보조 저장 장치

- 데이터 접근 시간이 메인 메모리에 비해 얼마나 느린가?
⇒ 약 ()
- 보조 저장 장치의 접근 횟수(number of disk I/O)가 프로그램 성능 평가 요소
⇒ 파일 구조 선정의 중요한 요소

❖ 파일 구조 선정 요소

◆ 파일 구조 선정 요소

- 가변성
- 활동성
- 사용빈도 수
- 응답 시간
- 파일 크기
- 파일 접근 유형

▶ 가변성(volatility)

◆ 파일의 성격

- 내용이 변하지 않는 정적 파일
- 내용이 자주 변하는 동적 파일

◆ 가변성(volatility)

- 전체 레코드 수에 대해 추가되거나 삭제되는 레코드 수
- **가변성이 높고 동적인** 파일은 빠르게 접근되고 갱신될 수 있도록 파일을 구성해야 함

▶ 활동성(activity)

◆ 파일의 활동성

- 주어진 기간 동안에 파일의 총 레코드 수에 대해 접근한 레코드 수의 비율
- 활동성이 **높으면** () 파일 구조가 좋을 수 있음

레코드

접근된 레코드/총 레코드 :

개 / 8개

접근된 레코드/총 레코드 :

개 / 8개

▶ 사용 빈도수 (frequency of use)

◆ 파일의 사용 빈도수

- 가변성과 활동성에 밀접히 관련
- 파일 설계에 중요한 요소가 됨
- 파일 사용이 빈번할수록 파일에 대한 빠르게 접근되고
갱신될 필요성은 더욱 높아질 것임

레코드

접근 횟수 : 회

접근 횟수 : 회

▶ 응답 시간(response time)

◆ 응답 시간과 파일 구조

- 검색이나 갱신에 대해 요구하는 자연 시간
- 파일에 대한 접근 방법을 결정하는데 중요한 요소가 됨
- 순차적으로 정렬된 키에 따라 레코드를 검색한다면 () 접근 방법으로 조직하는 것이 유리함
- 초 단위로 빠른 응답 시간을 필요로 하면 () 접근 방법을 선택해야 됨

▶ 파일 크기(file size)

◆ 파일 크기와 파일 구조

- 레코드 수와 각 레코드 길이가 파일 크기 결정
- 시간이 지남에 따라 파일 크기 성장(레코드 길이 확장, 레코드 수 증가)
- 처음 파일 생성할 때, 나중에 추가될 레코드들의 저장 공간을 예비해 두어야 함
- 성장을 유연하게 수용할 수 있는 파일 구조 필요
- 파일 성장을 유연하게 수용할 수 없을 때는 파일을 **재조직(reorganization)**해야 됨

▶ 파일 접근 유형

◆ 파일 접근 유형과 파일 구조

- 연산의 유형과 접근 형식에 따라 파일 구조 결정
- 연산 유형: 판독 위주 접근 or 갱신 위주 접근 ?
- 접근 형식: 순차 접근 주도 or 임의 접근 주도 ?