

데이터베이스 관리 시스템

문혜영

학습 목차

- 01 DBMS 개념
- 02 DBMS 장단점
- 03 DBMS 발전과정

01 DBMS 개념

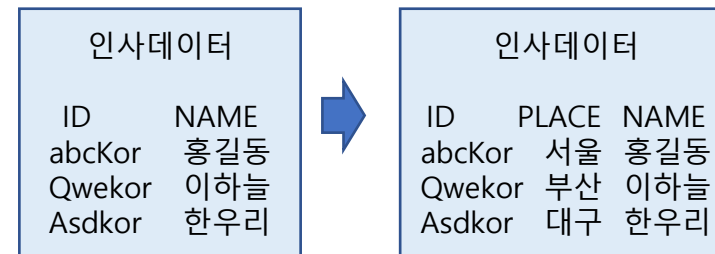
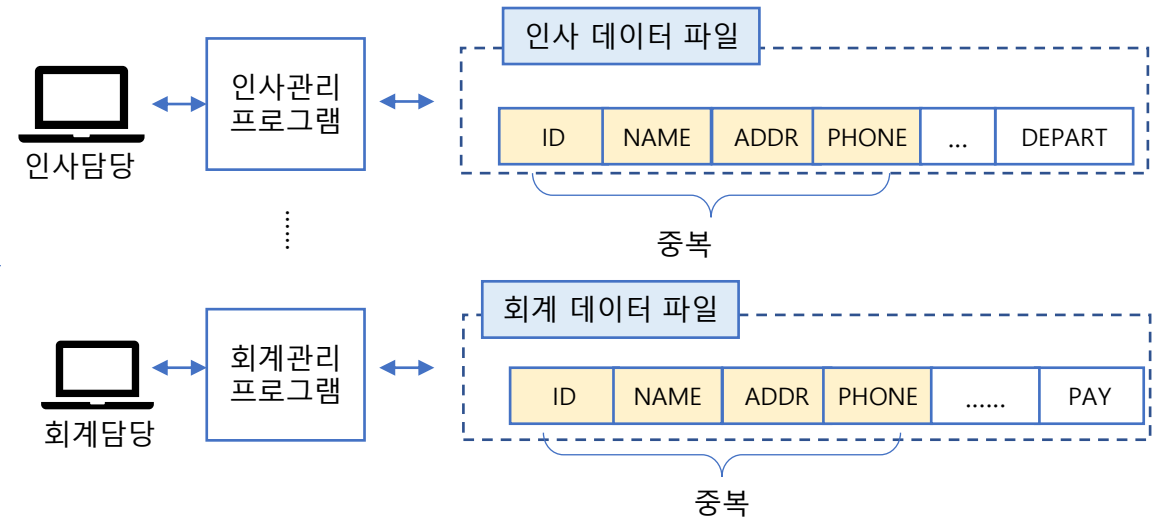
- 파일 시스템(File System)

- 중복 저장

- > 저장 공간 낭비,
데이터의 일관성, 무결성 유지 어려움

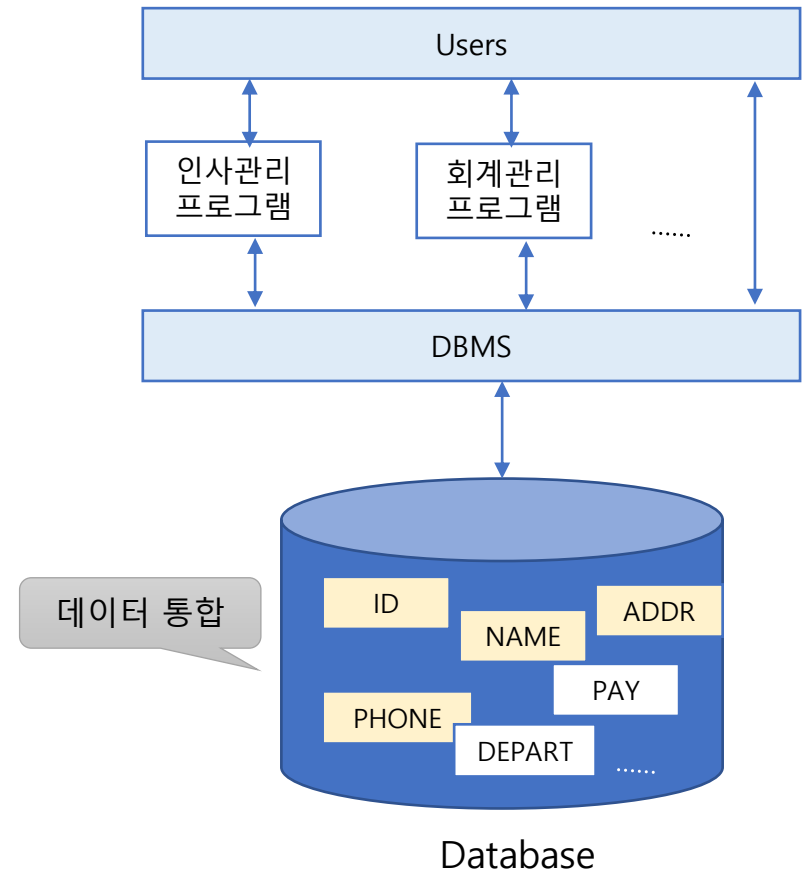
- 응용 프로그램이 데이터 파일에 종속적
(파일의 구조 변경 -> 응용 프로그램 변경)

- 병행 처리의 어려움



01 DBMS 개념

- **Database**
 - 데이터 및 데이터간 관계의 집합
- **데이터베이스 관리 시스템 DBMS(DataBase Management System)**
 - 사용자와 데이터베이스 사이에서 데이터베이스를 관리해주는 SW
 - 데이터를 데이터 베이스에 통합하여 저장하고 관리(중복의 최소화)
 - ✓ 중복성으로 인한 기존의 문제점
 - ✓ 일관성, 보안성, 경제성, 무결성
 - DB의 구조를 변경하여도 응용프로그램을 변경할 필요 없다(독립성)



01 DBMS 개념

- 질의(Query) 예

- "신규고용" 정책을 제안한 부서명? (검색)
- "총무과"에서 제안한 정책번호를 999로 변경하세요? (수정)
- "인사과"에서 제안한 정책명을 삭제하세요.(삭제)

직원		
직원 번호	이름	부서
10	김	B20
20	이	A10
30	박	A10
40	최	C30

부서	
부서 번호	부서명
A10	기획과
B20	인사과
C30	총무과

정책		
정책 번호	정책명	제안 직원
100	인력양성	40
200	주택자금	20
300	친절교육	10
400	성과금	30
500	신규고용	20

- "DB"를 수강한 학생 이름은?(검색)
- 학번이 300인 학생이 수강한 과목의 학점을 3으로 수정하시오.? (수정)

과목		
과목명	과목코드	학점
TEXT	T10	3
DB	D10	3
MATH	M10	2
KOR	K10	2

성적		
학번	과목코드	성적
100	T10	A
200	D10	C
300	M10	A
400	K10	A
500	M10	B

학생	
학번	이름
100	홍길동
200	진달래
300	보아라
400	이루리
500	무지개



- DBMS의 주요기능
 - 정의 기능(definition faculty)
 - 데이터베이스 구조에 대한 정의, 변경, 삭제(CREATE, ALTER, DROP)
 - 데이터의 형(Type), 제약조건 명시
 - 조작기능 (manipulation faculty)
 - 사용자와 데이터베이스 간의 인터페이스를 제공.
 - 데이터 삽입, 삭제, 수정, 검색(INSERT, DELETE, UPDATE, SELECT)
 - 제어기능 (control faculty)
 - 무결성 유지 되도록 제어(정확한 데이터 유지)
 - 보안유지, 권한검사(허가된 데이터만 접근)
 - 병행제어(여러 사용자 동시 접근 → 처리결과 정확성유지)
 - GRANT, REVOKE, COMMIT, ROLLBACK

```
click,close-full-overlay" data-bbox="700 0 975 115">review"),render:function() {var
ter,navigate(c.router,basename
$.addClass("iframe-ready")
removeClass("iframe-ready")
```

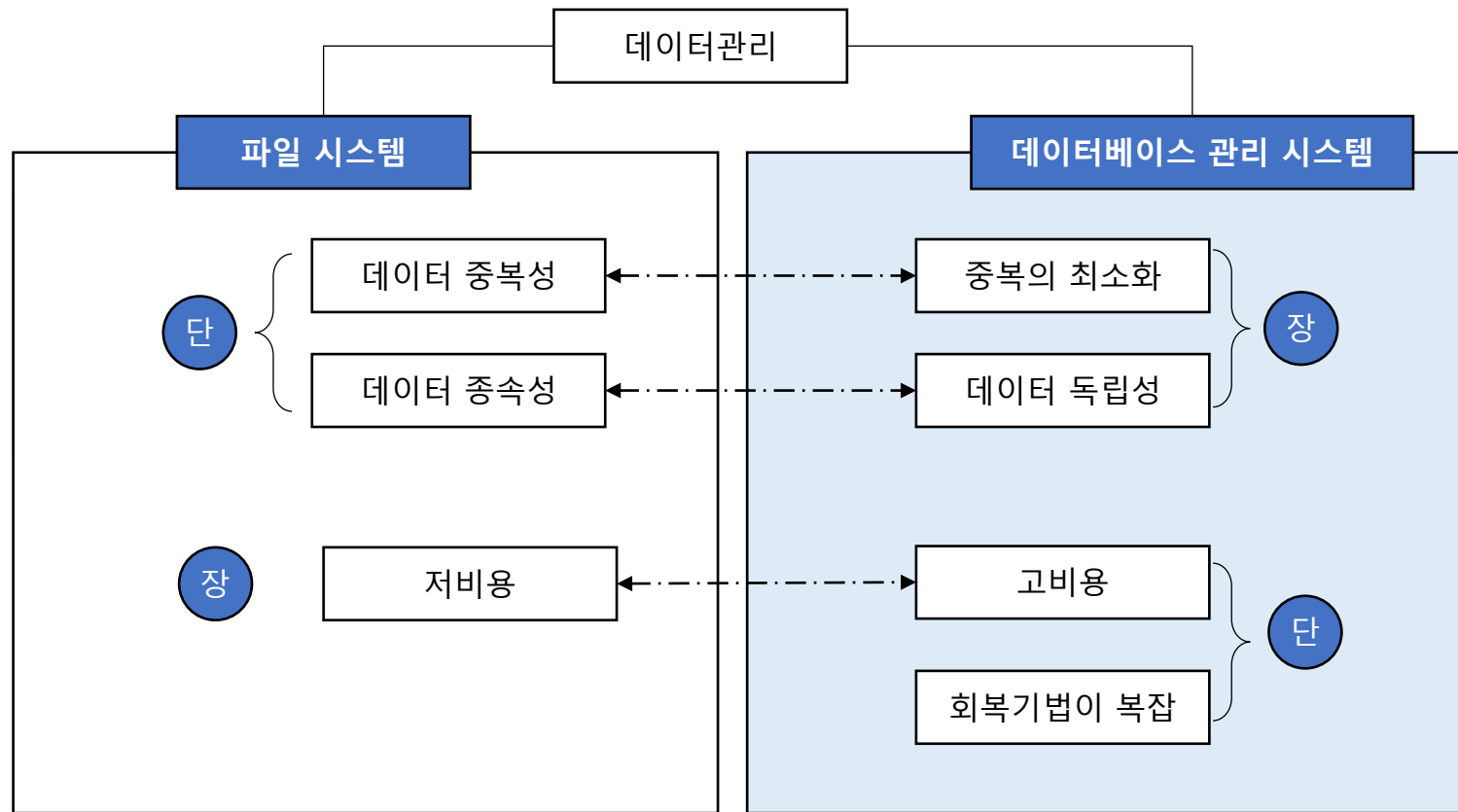
- 장점

- 데이터 중복 최소화
- 독립성 확보
- 동시 공유
- 보안 향상
- 무결성 유지
- 표준화
- 장애 발생시 회복기능
- 응용프로그램 개발 비용이 줄어 듦

- 단점

- 데이터베이스 구축 비용 증가
- 백업과 회복 절차가 복잡
- 중앙집중관리

02 DBMS 장단점



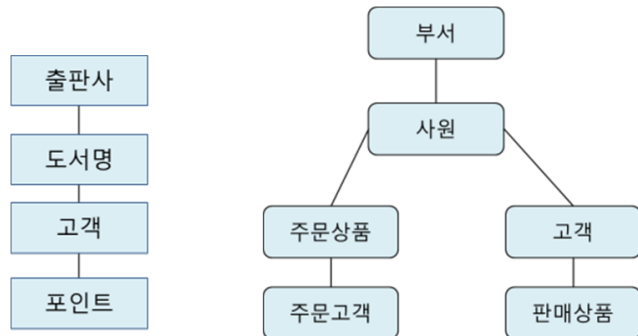

```
click,close-full-overview,review"),render:function(){$.fn.navigate(c.router,router,$.sel.addClass("iframe-embed"),$.fn.removeClass("iframe-embed"))}
```

1. 파일 시스템

- 각 응용프로그램마다 별도의 데이터 파일을 관리
- 데이터 중복성, 데이터 무결성 문제, 데이터의 물리적 및 논리적 종속성 등의 문제

2. 계층형 DBMS(hierarchical)

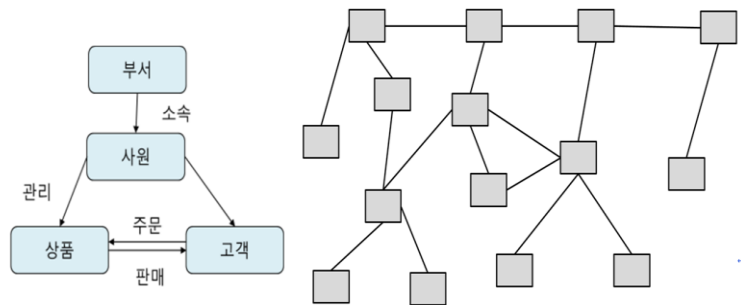
- 1960년대 후반, IBM의 IMS(Information Management System)에서 처음으로 사용
- 데이터를 계층 구조로 표현, 트리구조 사용하여 데이터간의 관계를 나타냄
- 1:N의 관계, 부모 자식 관계



```
back,close-full-overlay()<br>review"),render:function()<br>ter,navigate(c.router,basename<br>$.el.addClass("iframe-root")<br>removeClass("iframe-root")</pre>
```

3. 네트워크형 DBMS(Network)

- 1970년대, CODASYL (Conference on Data Systems Languages) 모델을 기반으로 발전.
- M:N 관계도 표현(계층형 모델보다 복잡성이 증가)
- 그래프 형태로 표현



```
click.close-full-overview()
review"),render:function() {
ter.navigate(c.router, {name:
$.addClass("iframe-right")
removeClass("iframe-right")
```

4. 관계형 DBMS (RDBMS, Relational Database Management System)

- 1970년, E.F. Codd에 의해 소개
- 데이터를 테이블(행, 열) 형식으로 저장
- SQL(Structured Query Language)을 사용하여 데이터를 조회, 추가, 수정, 삭제
- 현. 가장 널리 사용되는 DBMS 유형
- Oracle, MySQL, Microsoft SQL Server, PostgreSQL 등이 포함됩니다.

학번(SNO)	이름(SNAME)	학과(DEPT)	학과코드
101	홍지만	정보통신	001
102	정민기	정보통신	001
103	무뇌영	웹디자인	002
104	정수현	웹디자인	002
105	서유리	사이버안보	003

```
    .close-full-overview()
    .render(function() {
      router.navigate(c.router, {
        $el.addClass("iframe-container")
        removeClass("iframe-container")
      })
    })
  }
}
```

5. 객체 지향 DBMS((Object-Oriented Database Management System, OODBMS)

- 1980년대와 1990년대 초, 객체 지향 프로그래밍의 인기와 함께 개발
- 데이터를 객체로 표현하며, 상속, 캡슐화, 다형성과 같은 객체 지향 개념을 지원

6. NoSQL DBMS(Not Only SQL)

- 2000년대 이후, 관계형 모델의 구조적 제한 없이 대용량의 분산 데이터 저장을 지원(비 관계형데이터베이스)
- **분류**
 - 키-값 스토어 (Key-Value Stores)
 - 문서기반 데이터베이스(Document-Oriented Databases)
 - 칼럼 지향 데이터베이스(Columnar Databases)
 - 그래프 데이터베이스(Graph Databases)
- 예) Redis, MongoDB, Apache Cassandra , Neo4j 등

```
click.close-full-overview()
review"),render:function()
ter.navigate(c.router,back
$.el.addClass("iframe-resize")
removeClass("iframe-resize")
```

7. NewSQL DBMS:

- 관계형 DBMS의 SQL 표준 쿼리 기능을 유지하면서 대규모 분산 환경에서의 확장성을 제공하려는 시도
- Google Spanner, CockroachDB 등



- 파일시스템의 문제점과
데이터베이스 관리 시스템의 장점에 대해 설명 하세요.

다음에서 설명하는 파일 시스템의 문제점은?

“파일의 레코드 크기나 타입이 변경되면

해당 파일에 액세스 하는 프로그램 코드도 수정되어야 한다. ”

- ① 데이터 중복성
- ② 데이터 독립성
- ③ 데이터 종속성
- ④ 데이터 재사용성



• 다음 중 데이터베이스 관리 시스템의 장점으로 보기 어려운 것은?

- ① 데이터 보안 향상
- ② 데이터 중복 최소화
- ③ 적은 비용
- ④ 데이터 독립성



수고하셨습니다 🙌