

20240306 수업 정리

파일 시스템으로 구축된 구매 및 판매 응용 프로그램에 관한 설명 (2p)

- 하나의 데이터베이스가 여러 사람에 의해 사용될 때,
데이터 조작으로 인해 정합성과 무결성 문제 발생 가능성이 있다.
- 파일 시스템에서 구매와 판매 담당이 동시에 파일을 수정하면 **정확성 기준이 사라지는** 문제가 발생한다.
- 과거 파일 시스템에서 만약 A라는 사람이 데이터를 수정하고 작업을 마치면,
오류를 발견해도
이전 값 추적이 어려워 정확한 값을 찾기 어렵다.
- 결국 데이터 값에 오류가 생겨도 **원인 추적이 어려워** 큰 문제로 이어질 수 있다.

→ 이러한 문제들을 해결하기 위해서 DBMS를 사용하여 데이터 정확성을 유지하고자 한다.

- DBMS로 인한 **데이터 변경** 문제 예방 / **권한 부여**로 데이터 접근 제어

데이터 관리자는 원천 데이터까지 접근 가능하다

- **데이터 거버넌스**팀이 데이터 접근 및 관리 규제, 대기업 대부분이 이런 방식으로 데이터를 관리한다.



그림 1-20 파일 시스템으로 구축된 구매 및 판매 응용 프로그램

DBMS로 구축된 구매 및 판매 응용 프로그램에 관한 설명 (2p)

- 데이터 분석에서 사용되는 응용 프로그램은 SQL, Oracle, Tableau, Power bi 등이 있으며,
대시보드로 데이터를 시각화하여 분석한다.
- 데이터 관리자들은 대시보드를 통해 데이터를 파악하지만, **원천 데이터를 수정하지 않는다**.
- 구매, 판매 담당자가 각자 필요로 하는 툴을 통해 데이터를 관리하고,
데이터 관리자는 데이터에 대한 권한을 갖는다.
- 데이터 정합성과 정확성 유지를 위해 시스템을 구축하여 문제를 예방하고 해결한다.

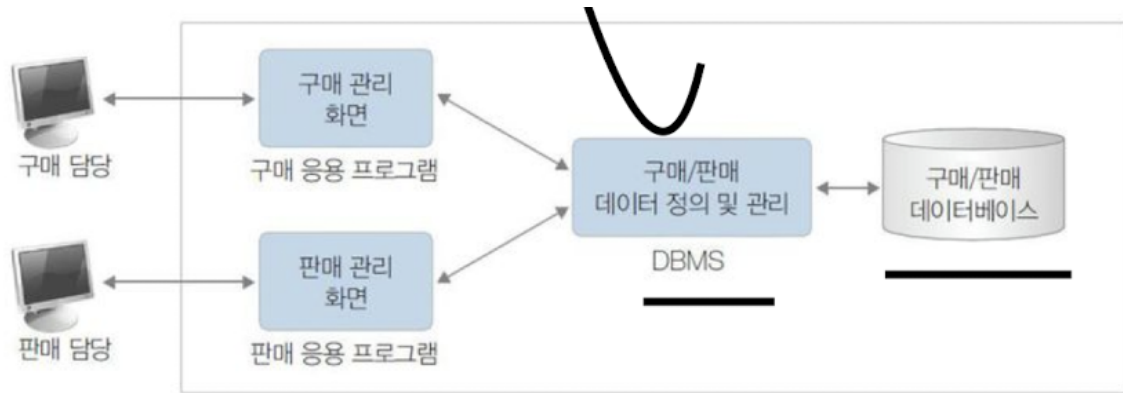
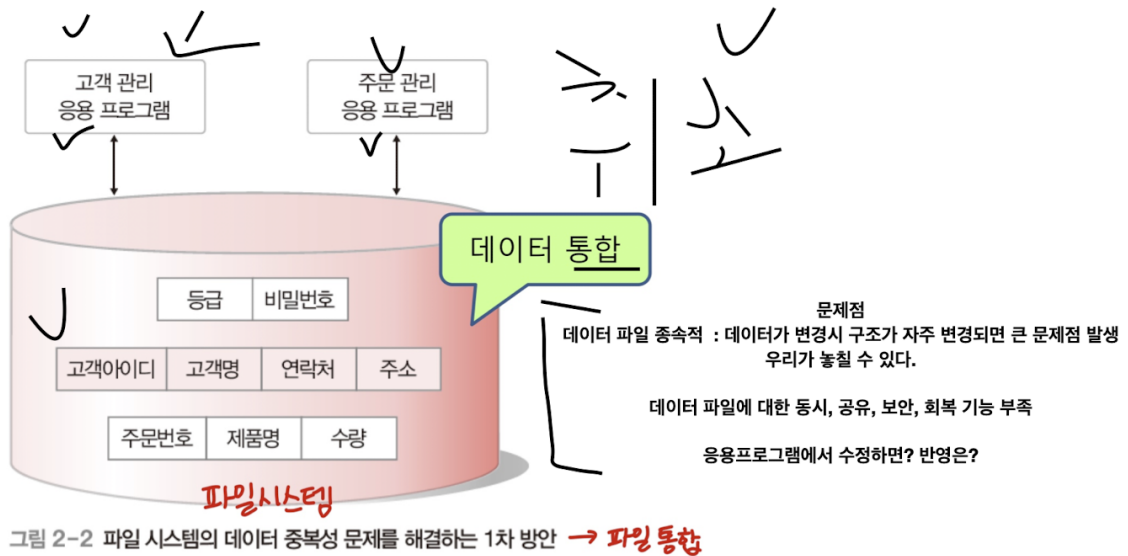


그림 1-21 DBMS로 구축된 구매 및 판매 응용 프로그램

- 데이터 관리 시스템은 고객 정보 **암호화**와 **비식별화**를 통해 보호하며, 데이터베이스 자체에서는 고객의 개인정보 절대 사용이 금지된다.
- 악의적인 해커나 정보 유출로부터 보호하기 위해 **내부적인 검증과 제어** 중요하며, 데이터 수정 가능성으로 발생하는 정확성 문제를 해결하기 위해 **리더 베이스 관리 시스템**이 도입되었다.
- 데이터에 대한 **동시 공유, 보완, 회복** 기능은 여전히 필요하며, 데이터베이스 관리 시스템을 통해 데이터에 대한 효율적인 관리가 이뤄지고 있다.
- 데이터 관리 시스템을 통해 **제한된 접근 권한**을 가진 관리자만 데이터를 조작할 수 있고, 다른 부서는 데이터를 건드릴 수 없도록 되어 있다.

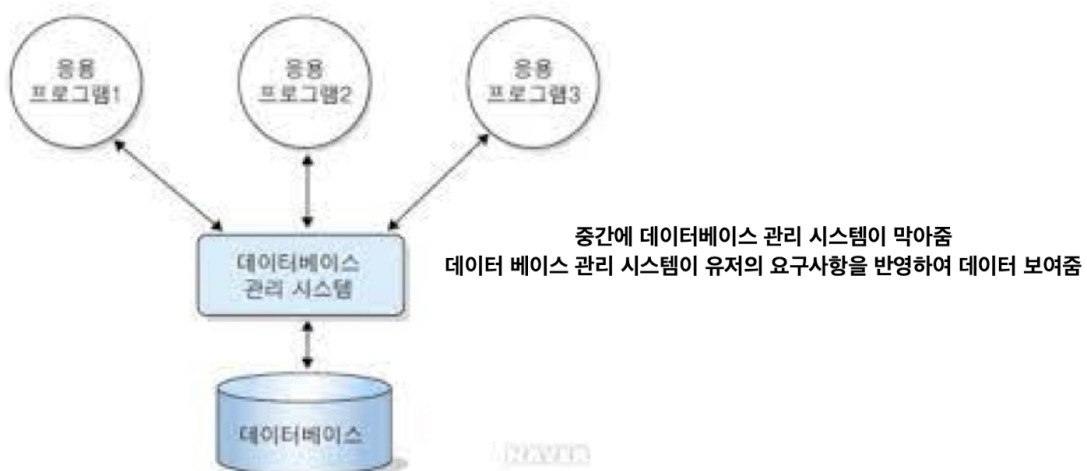
파일 시스템의 데이터 관리의 중요성 (3p)

- 파일 시스템의 통합을 처음으로 시도했지만 **데이터 수정**으로 발생한 정확성 문제가 생겨 데이터 관리의 중요성을 인지하였다.
- 데이터베이스를 관리하는 관리자는 데이터를 수정할 수 있지만, 실수로 데이터를 잘못 건드릴 경우 예기치 못한 결과가 발생할 수 있다.
- 이러한 문제 발생 시, 관리자는 **데이터 복구**와 **에러 해결**의 역할을 담당하며, 이는 엔지니어와 데이터 전문가가 주로 수행하는 작업이라고 한다.
- 데이터 관리에 대한 이해가 있는 사람들은 이러한 일을 처리할 수 있지만, 그렇지 않은 사람들이 데이터를 수정하거나 삭제하는 행위는 막아야 한다.



기업의 데이터 접근 권한 (5p)

- 기업에서 데이터를 통해 부서들에 인사이트 제공이 중요하나, 데이터에 접근하기 어려운 다른 팀들이 있다.
- 전체 권한이 있는 데이터팀은 쉽게 데이터에 접근하고 다양한 시스템을 관리, 다른 팀들은 권한으로 제한된다.
- 기업에서는 권한으로 인해 데이터 분석이 어려울 수 있으며, 권한 설정이 효율적인 업무변화를 이끌 수 있다.
- 데이터베이스 관리 시스템과 SQL을 통한 데이터 추출과 분석, BI툴 활용이 중요하며 다양한 시스템을 이해하는 것이 필요하다.



정의 기능 - 데이터베이스 기본 구조와 테이블 설계 (6p)

- 정의, 조작, 제어 기능을 이용해 데이터베이스 구조를 만들고 추후 데이터를 가공하거나 새로운 데이터를 생성한다.

- SQL에서 DCL, DML, DDL을 활용하여 테이블을 만들고 데이터베이스를 설계하는 과정을 거친다.
- 고객 아이디, 주문 번호, 휴대전화번호, 주소지 등과 같은 데이터가 열과 행 형식으로 구성된 테이블을 설계한다.
- 스키마 정보 확인을 통해 데이터 타입, 길이, 랭크 등을 결정하며, 문자열과 숫자로 이뤄진 데이터 처리 방법에 대해 고민한다.

정의 기능 - 데이터베이스 설계 (6p)

- 데이터베이스 설계는 **Excel 테이블과 같은 형태로** 이뤄지며, 각 항목을 고민하고 설계해야 한다.
- 데이터에 대한 **식별자 값**이나 **포인트키**에 대한 고민이 필요하며, 상품정보와 같은 사항을 고려할 때 그 역할을 고민해야 한다.
- 데이터베이스의 경우 **보안 서버**에 저장되며, 민감한 정보는 가려져 있어야 한다.

조작 기능(SQL) (6p)

- SQL은 데이터를 수정하거나 삭제하거나 추가하는 등 다양한 조작 기능을 수행하는데, 주로 'select', 'from', 'where' 등의 문법을 통해 이루어진다.
- 쿼리를 통해 데이터베이스를 관리하고 조작할 때, SQL문법을 잘 숙지하면 많은 일을 할 수 있다는 점이 중요하다.
- 즉, SQL을 사용하여 데이터를 효율적으로 관리하고 활용할 수 있어야 한다.

제어 기능 - 데이터베이스 중복 처리 (6p)

- 데이터베이스는 관리 시스템으로 중복 처리가 필요하며, 중복 데이터는 허용되지 않으며 설계가 중요하다.
- 마스터 테이블 등 중복이 절대 허용되지 않는 테이블이 중요하다.
- 중복으로 인한 문제는 응용프로그램 전반에 영향을 미친다.
- 잘못된 데이터는 전체 시스템에 영향을 줄 수 있어 중복을 철저히 제거해야 한다.

제어 기능 - 데이터 보안과 권한 (6p)

- 데이터 원천 수정이 필요한데 중요한 점은 데이터 보안과 무결성을 유지해야 한다.
- 권한을 통한 데이터 접근 제한은 중요하며, 데이터베이스에 대한 이해로 중복 및 로직 파악이 필요하다.
- 권한에 따라 데이터 접근이 다르고 책임도 증가되며, 데이터베이스 이해 없이 데이터 검증시 문제 발생 가능성이 있다.

데이터베이스 장점 (7p)

독립성

데이터베이스 관리 시스템과 응용프로그램이 독립적으로 작동해야 한다.

데이터 동시 공유

데이터 관리팀의 역할이 중요하며, 다양한 팀에서 데이터를 다르게 해석하고 활용한다.

데이터 보안 향상

다른 팀에서 데이터베이스를 조작하거나 수정하는 권한은 보안 시스템을 통해 제한되어야 한다. 또한, 내부에서

보안 유출을 막기 위해 협의를 통해 권한을 부여하는 조직이 중요하며, 정확한 데이터는 필수이다.

데이터 무결성 유지

데이터의 무결성은 데이터의 정확성을 의미하며, 정확성이 맞지 않으면 심각한 문제가 발생할 수 있다.

데이터 요청 시 반드시 데이터의 정확성을 검증해야하며, 이는 데이터 처리에서 중요한 역할을 한다.

표준화

데이터 표준화는 기업 입장에서 데이터를 표준화하여 전사 차원에서 바라볼 수 있게 하며, 예를 들어 주문 금액이라는 기준이 있을 때 주문 금액의 다양한 해석(ex) 마케팅과 영업)으로 인한 어려움 해소를 위해 정확한 기준 정리의 중요성이 강조된다.

장애발생시 회복

데이터의 관리는 장애 발생 시의 중요성을 강조하며, 예전과 달리 관리 시스템을 통해 데이터 손실을 최소화하고 히스토리를 관리하여 복구가 가능하다.

데이터베이스 단점 (7p)

고비용 / 백업과 회복 방법 복잡

과금체계와 권한 관리(권한에 따라 접속할 수 있는 것이 다르기에)에서 발생하는 이슈로, 기업 입장에서 고려해야 하는 부분이다.

중앙 집중화 관리로 인한 취약점

데이터베이스 관리를 하나의 부서에서 중앙집권화하면 보안상 취약점이 존재할 수 있다.

공격을 받을시 중앙화된 시스템은 전체가 위협에 노출되는 점을 주의해야 한다.

이에 대한 방안으로 데이터베이스 관리를 분할하면 중앙집권화에 대한 문제를 완화할 수 있다.

데이터베이스 관리 시스템 발전 과정

1세대 데이터베이스 관리 시스템: 네트워크 계층 DBMS

- 예를 들어 고객, 직원, 주문, 상품, 판매라는 항목이 있을 때 이 항목들 간의 관계를 통해 이뤄지는 것들을 네트워크처럼 표현한 것이다.

- 계층적 구조로는 데이터 관리에 문제가 있고, 유연한 대응과 불편함을 겪는다.

- 이에 대한 해결책으로 나온 것이 관계형 DBMS이다.

2세대 데이터베이스 관리 시스템: 관계 DBMS

- 관계형 DB는 행과 열의 개념으로, 주로 MySQL이나 Oracle과 같은 DBMS를 사용한다.

3세대 데이터베이스 관리 시스템: 객체지향, 객체관계 DBMS

4세대 이후 데이터베이스 관리 시스템: NoSQL, NewSQL DBMS

- 주로 비정형데이터를 관리하는데 사용한다.