



3 / 13

- 가장 중요한 요소, Entity, Attribute, Relationship
 - Entity
 - 엔터티, 객체
 - Attribute
 - 특성과 속성, 필드 혹은 컬럼으로 불린다.
 - Relationship
 - 엔터티 사이의 관계
 - 1 to many
 - many to many
 - 1 to 1
- E-R Diagram
 - 데이터베이스를 시각적으로 표현하는데 사용
 - 직사각형은 엔터티이다.
 - 엔터티들 사이의 라인들은 관계를 나타낸다.
 - 각각의 회사마다, 다를 수 있기 때문에 여러 버전을 사용해 봐야한다.
- Many to Many의 관계에서 꼬일 경우, 따로 빼서 1 to Many로 간다.
- DBMS
 - 사용자가 데이터베이스와 소통하게 해주는 프로그램
 - Access, Oracle, mysql
 - TAL Distiributors는 Access를 사용

- Database Design
 - 필요한 데이터 베이스의 구조를 결정하는 작업
- 2-tier Architecture
 - Web Server - DBServer 두개의 경우
- 3-tier Architecture
 - Web Server - WAS Server - DBServer 3개의 경우
- Forms
 - 마이크로 스프트 액세스에 국한된 애기
 - 데이터베이스로 부터 데이터를 과리하고 프린트하기위한 스크린 객체
 - DBMS는 TAL Distributors가 필요로하는 Form들을 만들도록 도와준다.
 - 유저들은 이 폼들을 이용해 데이터를 입력
 - EX) 새로운 아이টে을 입력, 보고, 바꾸고, 지울 수 있다. enter, view, change, delete
 - Report - 보완
 - **사용자의 needs에 맞게**
- 데이터베이스 사용에 장점
 - 같은 양의 데이터로 부터, 더 많은 정보를 얻어낼 수 있다.
 - 데이터를 공유할 수 있다.
 - 조직전체의 필요성에 더 부합
 - 중복을 컨트롤 한다.
 - **보완**
 - 완성성 / 무결성을 향상 (참조 무결성)
 - 무결성 제약 - 데이터베이스에서 데이터가 따라야 하는 룰
 - 보안 향상
 - 생산성 향상

- 데이터의 독립성 제공
 - 데이터베이스를 액세스하는 프로그램의 변화 없이 데이터베이스의 구조를 변화시킬 수 있다.
- 데이터베이스 프로세싱의 단점
 - 데이터베이스 파일의 크기가 커지고, 메모리를 많이 차지한다.
 - DBMS의 복잡한 feature들을 이해하는 노력이 필요
 - 파일 시스템 처럼 분리 되어있지 않아, 한 유저 시스템의 실패가 다른 곳에도 영향을 줄 수 있음
 - 파일 시스템 보다 복잡한 구조, 동시 사용자가 많음

Next.. 케이스 소개 및, 액세스를 사용을 해야한다. 늦어도 다음주 까지, Install하