**컴활 3과목**

**1강) 데이터베이스 개요**

**데이터베이스 정의**

* DB의 도입배경
* 독립된 파일 단위(여러 다른 프로그램 및 확장자로) 저장하게 되면 데이터 종속성 및 중복성이 높아 무결성 위배 가능성이 높아짐. (쉽게 말해, 어디는 내가 20살이고 어디는 22살이고 이런식으로 다른 정보가 기입될 수가 있음. 멀티버스마냥)   
  이러한 단점을 보완하고자 여러 시스템이 공용으로 데이터를 모아 관리하는 데이터베이스(DB)를 구축하게 됨.
* DB의 장단점
* 장점 :

1. 각 프로그램으로 문제를 수정하지 않아도 되어 **종속성이 최소화** 됨.
2. 정보가 데이터베이스에 한 번만 저장되므로 **중복성이 최소화** 됨.
3. 따라서 **일관성**이 유지되며,
4. **무결성**이 유지됨.
5. 하나의 공간에서 작업하므로 **공유가 쉽고**,
6. 접근 권한을 이용해 **보안성을 강화**할 수 있음.
7. 각 부서에서 처리한 내용이 실시간으로 데이터베이스에 입력되므로 **실시간 처리가 가능**함.

* 단점 :

1. 전문가가 부족함.
2. 데이터 베이스를 구축/관리할, 전산화 비용이 증가함.
3. 시스템이 복잡하고 느림.
4. 파일 회복이 어려움.

**데이터베이스 관리 시스템(DBMS)**

* 텍스트, 원, 스크린샷, 로고이(가) 표시된 사진

  자동 생성된 설명사용자와 DB 사이에서 사용자의 요구에 따라 DB를 생성하거나 관리해주는 소프트웨어를 말함.
* 사용자 <-> DBMS <-> DB
* 사용자 : DBA(DB관리자), 일반 사용자, 응용 프로그램
* DBMS의 종류 : 계층형, 네트워크형, 객체지향형, **관계형**

텍스트, 폰트, 로고, 그래픽이(가) 표시된 사진

자동 생성된 설명

위 사진 같은 DBMS들이 존재함.

**DBMS 기능 및 언어**

1. 정의어(DDL : **D**ata **D**efinition **L**anguage) (DBA 사용)  
   DB를 생성하거나 자료 형태(type)와 구조 등을 수정하며 데이터를 이용하는 방식을 정의하는 기능.  
   CREATE(엑세스에서 테이블을 만들 때 사용), ALTER(테이블의 구조를 바꿀 때), DROP(테이블을 지울 때)
2. 조작어(DML : **D**ata **M**anipulation **L**anguage) (사용자 사용)  
   데이터의 검색, 삽입, 삭제, 변경 등을 처리하는 기능  
   SELECT(검색할 때), INSERT(삽입할 때), DELETE(삭제할 때), UPDATE(변경할 때)
3. 제어어(DCL : **D**ata **C**ontrol **L**anguage) (DBA 사용)  
   데이터의 무결성을 유지하기 위한 언어. 데이터의 무결성을 유지하기 위한 보안 및 권한 검사, 병행 제어 등의 기능을 정의하는 기능. (관리를 목적으로 이용되는 언어.)  
   COMMIT(작업을 승인), ROLLBACK(다시 되돌림), GRANT(권한 부여), REVOKE(권한 빼앗음)

**관계형(Relational-DBMS = RDBMS)**

* 테이블(Table)을 이용한 구조

텍스트, 폰트, 번호, 스크린샷이(가) 표시된 사진

자동 생성된 설명

* 테이블 : 표, 개체 또는 **릴레이션**
* 필드 : ==속성(Attribute), 열
* 레코드 : ==튜플(Tuple), 행
* 도메인 : 하나의 속성에서 취할 수 있는 원자값의 범위.

텍스트, 번호이(가) 표시된 사진

자동 생성된 설명

* 필드(속성)의 개수 : 차수(Degree)  
  텍스트, 폰트, 번호, 스크린샷이(가) 표시된 사진

  자동 생성된 설명
* 레코드(튜플)의 개수 : 기수(Cardinality)  
  텍스트, 번호, 폰트, 스크린샷이(가) 표시된 사진

  자동 생성된 설명

**테이블의 특징**

* 속성(필드)과 튜플(레코드) 들은 유일하며, 순서가 무관함.
* 속성(필드)의 값은 분해 할 수 없다. (각각의 도메인들은 소립자처럼 더 이상 분해 불가능한 단위이다.)
* 속성(필드)의 값은 동일할 수 있다.
* 튜플(레코드)은 삽입, 삭제 등에 의해 계속 변한다.
* 튜플(레코드)를 식별하기 위해 속성(필드)의 일부를 Key로 설정한다.

**2강) 데이터베이스 설계**

**키(KEY)**



* 후보키(Candidate Key) : (학번, 주민번호)
* 유일성, 최소성이 모두 만족되는 속성.
* 기본키(Primary Key) : (학번 또는 주민번호)
* 후보키 중 선정(중복 X, Null X <- 개체 무결성)
* 대체키(Alternate Key) : 기본키가 “학번”이라면, “주민번호” 라는 후속 키를 말함.)
* 기본키가 아닌 후보키.
* 슈퍼키(Super Key) : (“이름” + “주소”)
* 두 개 이상의 속성을 연결하여 유일성 만족. (두 개 이상의 속성이 전부 겹치지 않을 경우만 체크함.)
* 유일성 만족, 최소성X
* 외래키(Foreign Key)
* 관계가 정의되어 참조된 다른 테이블의 기본키.

**데이터베이스의 설계**

* 1단계 – 개념적 설계
* 요구사항을 분석하여 E-R D(Entity-Relationship Diagram)와 같은 개념적 구조 설계

도표, 텍스트, 라인, 폰트이(가) 표시된 사진

자동 생성된 설명

* 2단계 – 논리적 설계
* DBMS에 적합한 테이블의 구조로 설계

텍스트, 스크린샷, 폰트, 문서이(가) 표시된 사진

자동 생성된 설명

* 3단계 – 물리적 설계
* DBMS로 구현 가능한 물리적 구조(하드웨어) 설계

**E-R D ( 개체(Entity) – 관계(Relationship) 다이어그램 )**

텍스트, 도표, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

사각혈 – 개체(Entity) 타입

마름모 – 관계(Relationship) 타입

타원 – 속성(Attribute) 타입

밑줄 – 기본키

**데이터베이스 설계에서 논리적 설계를 못하면 생기는 일!!**

1. 삽입 이상(ㅗㅜㅑ) : 불필요한 데이터가 함께 삽입되는 현상

텍스트, 스크린샷, 폰트, 도표이(가) 표시된 사진

자동 생성된 설명

수강과목, 담당교수만 추가하니 이름, 나이 추가가 애매해짐.

1. 삭제 이상 : 필요한 데이터가 함께 삭제되는 현상

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

이배찌 학생의 정보만 제거하고 싶었는데, 담당교수 정보까지 지워져버림.

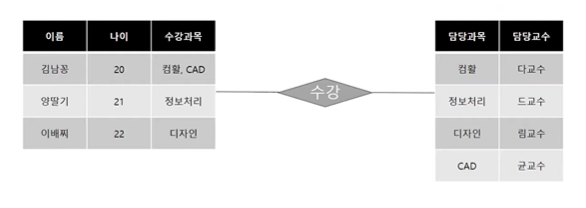
1. 갱신 이상 : 일부만 수정되어 데이터의 불일치 발생

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

1행의 김남꽁은 수정되지 않아 30이고, 4행의 김남꽁만 수정되어 20이다. 따라서 데이터가 불일치하게 됨.

해결방법 :



이런식으로 분리해주면 된다.

더 작은 필드를 갖는 테이블로 분해하는 과정.

이게 바로 **정규화**라고 한다.

**정규화**

* 이상(Anomaly) 현상이 일어나지 않도록 분해하는 과정
* 중복, **종속성을 최소화**하기 위한 작업
* 논리적 설계 단계에서 수행해야 함.
* 속성 수가 적은 테이블로 분할되어 관리가 쉬워짐. (유지보수 편리)
* 분할이 너무 많으면 데이터 연결이 복잡해져 오히려 효율이 떨어짐.
* 완벽하게 중복을 제거할 수는 없음.

**데이터베이스의 구조(스키마)**

1. 외부스키마 : **일반** 사용자나 응용프로그래머의 관점에서 본 스키마(서브 스키마 = 사용자 뷰)
2. 개념스키마 : DB의 전체적인 **논리**적 구조이며 접근 권한, 보안정책, 무결성 규칙에 대한 정의(스키마)
3. 내부스키마 : DB의 전체적인 **물리**적 구조 시스템 설계자의 관점에서 본 스키마(물리 스키마)

스크린샷, 도표, 디자인, 일러스트레이션이(가) 표시된 사진

자동 생성된 설명

**3강) 테이블 디자인**

**테이블 보기**

* [디자인 보기] : 테이블의 구조 및 필드 속성 변경

텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명

* [데이터시트 보기] : 레코드 입력 및 수정

텍스트, 스크린샷, 번호이(가) 표시된 사진

자동 생성된 설명

**데이터 형식**

1. 짧은 텍스트 : 텍스트 + 숫자 = 255자
2. 긴 텍스트 : 텍스트 + 숫자 = 63999자
3. 숫자 :
4. 바이트 : = 1 byte
5. 정수 : = 2 byte
6. 정수(Long) : = 4 byte
7. 실수(Single) : = 4 byte
8. 실수(Double) : = 8 byte
9. 날짜/시간 : 100년 ~ 9999년까지 입력가능 = 8 byte
10. 통화 : 정수 15, 소수 4자리까지 입력가능 = 8 byte
11. 일련 번호 :

* 자동으로 번호가 부여 됨.
* 데이터가 입력된 후에는 선택 불가능.
* 삭제된 일련 번호는 다시 부여되지 않음.
* 정수(Long) = 4 byte, 복제 ID = 16 byte

1. Yes/No : Yes/No, True/False, On/Off 선택 가능 = 1 bit
2. OLE 개체 : 개체 연결 포함, 기본키 설정 불가능. 이미지 파일 같은거 추가 가능. = 1 GB
3. 하이퍼링크 : 주소 데이터 입력 = 2048자
4. 첨부 파일 : 이메일에 파일을 첨부하는 것처럼 파일 첨부. 기본키 설정 불가능.

**조회 속성**

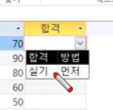
* 짧은 텍스트, 숫자, Yes/No만 설정 가능.
* 컨트롤 표시에서, 콤보상자(드롭다운) 및 목록상자(선택지) 적용 가능.
* 그 외 조회 속성 :

1. 행 원본 유형 : 테이블/쿼리, 값 목록, 필드 목록  
   테이블/쿼리로 설정하면 존재하는 테이블의 값을 받아와 사용할 수 있고,  
   값 목록으로 설정하면, 값을 직접 입력하여 목록을 만들 수 있다.  
   필드 목록은 필드의 값을 받아오는 듯.

행 원본 : 행 원본 유형의 값 입력. 세미콜론으로 구분.

1. 바운드 열 : 실제로 선택 시 저장되는 열. (1~)

열 개수 : 열의 개수를 지정함.



열 이름 : 행 원본 유형을 테이블/쿼리로 설정하였을 때, 열 이름을 예(YES)로 설정하면, 행 원본에 설정된 테이블 등에 따라 열 이름이 설정된다.

1. 열 너비 : 열의 너비를 설정한다. 1;2 이런식으로 지정하면, 1(세로)x2(가로) 크기의 열로 지정됨.
2. 목록 너비 : 너비의 총량. 전체. 5cm로 설정하고, 열 너비가 1;2라면, 2cm만큼의 열이 생기고, 두번째 열은 5-2 = 3cm로 생성됨.  
   텍스트, 스크린샷, 라인, 평행이(가) 표시된 사진

   자동 생성된 설명
3. 행 수 : 목록의 행 개수 설정. 전체 목록이 10개인데, 행 수를 3으로 한다면, 노출되는 행 수가 3으로 제한되고, 스크롤 바가 나타남.  
   텍스트, 번호, 라인, 폰트이(가) 표시된 사진

   자동 생성된 설명
4. 목록 값만 허용 : 값을 입력했을 때, 목록에 존재하는 값만 허용할지, 목록에 없는 값도 허용할지 정하는 boolean값.

**4강) 필드 속성**

**필드 속성**

1. 필드 크기 : 그냥 필드 크기. ㅋㅋㅋ
2. 형식 : 엑셀의 표시형식?과 같음.
3. 캡션 : 데이터시트에서 보여지는 레이블(제목)이 변경됨.
4. 기본값 : 자동으로 입력되어 있는 기본값. 초기화 하는 느낌인듯.
5. 필수 : 필수로 입력하는 속성. 안 입력하면 다음으로 못 넘어감. (띄어쓰기 하나만 넣고 넘어가려해도 안됨.)
6. 빈 문자열 허용 : 이걸 허용하면, 필수 설정된 칸에 띄어쓰기 하나만 넣고 넘어가도 허용됨.
7. IME 모드 : 자료 입력 형태(한글, 영어)를 설정해 한영키를 누르지 않고도 편리하게 전환할 수 있게 해 줌.
8. 소수 자릿수 : 소수점 이하 자릿수를 결정해줌.

**유효성 검사**

* 입력된 데이터가 유효한지 검사하는 기능
* ‘유효성 검사 텍스트’를 통해 오류 메시지 표현 가능

**인덱스**

* ‘중복 가능’, ‘중복 불가능’ 등을 지정 가능
* **여러 필드**에 설정 가능
* OLE, 첨부 파일 형식은 인덱스 설정 불가능
* 정렬 및 검색 속도가 빨라지지만, 전체적인 업데이트 속도가 느려짐.

**입력 마스크**

* 정확한 데이터가 입력될 수 있도록 해주는 ‘틀’
* 입력마스크 ; 저장방법 ; 표시문자  
  텍스트, 번호, 폰트, 스크린샷이(가) 표시된 사진

  자동 생성된 설명

**5강) 관계 정의**

관계 편집

* 항상 참조 무결성 유지 :   
  외래키 값은 참조된 테이블의 필드의 값과 동일해야 한다.  
  \*외래키란? 참조된 다른 테이블의 기본 키를 의미함. (구매 테이블의 과일 필드 같은.)
* 관련 필드 모두 업데이트 :   
  참조된 필드 값이 변경되면 자동으로 외래키 값도 변경됨. 그냥 원래 변숫값 바꾸면 변수를 사용하던 곳의 값도 변경되게 되는 느낌. 포인터마냥. 두 값을 연결함.
* 관련 레코드 모두 삭제 :   
  참조된 필드 값이 삭제되면 자동으로 외래키 값 삭제.