



데이터베이스

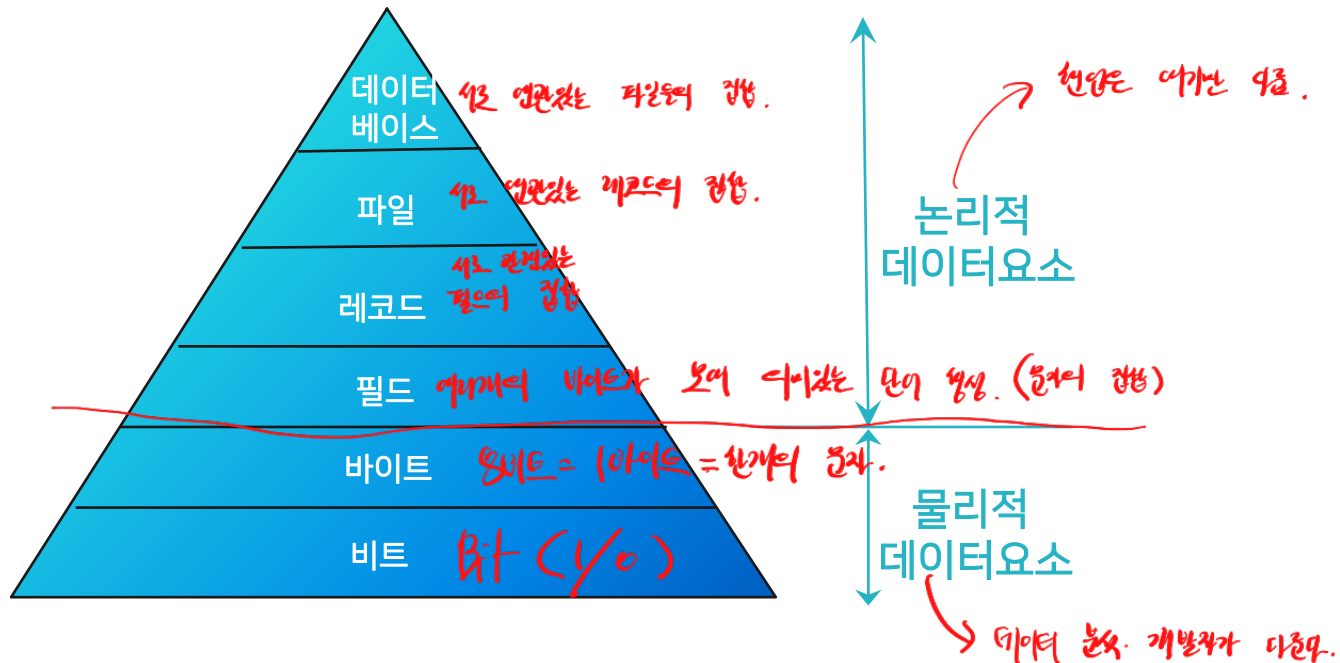
한은정 교수



학습 목표

1. 컴퓨터 데이터의 계층구조와 구성요소를 살펴본다
2. 기업 데이터를 구조적으로 설계하기 위한 데이터 모델링 기법에 대해 알아본다
① 개념적 ② 논리적
3. 개념적 데이터모델과 관계형 데이터베이스의 개념을 이해하고, 데이터 구조화 방법을 살펴본다
4. 기업의 데이터베이스 시스템에서 DBMS의 역할을 이해한다

데이터의 계층구조



데이터의 계층구조

- 물리적 계층

- 비트(bit): 데이터를 구성하는 가장 작은 단위 (0 또는 1)
- 바이트(byte): 8비트가 모여 1바이트를 이루며, 하나의 문자를 표현한다 (예: A, 1, \$ 등)

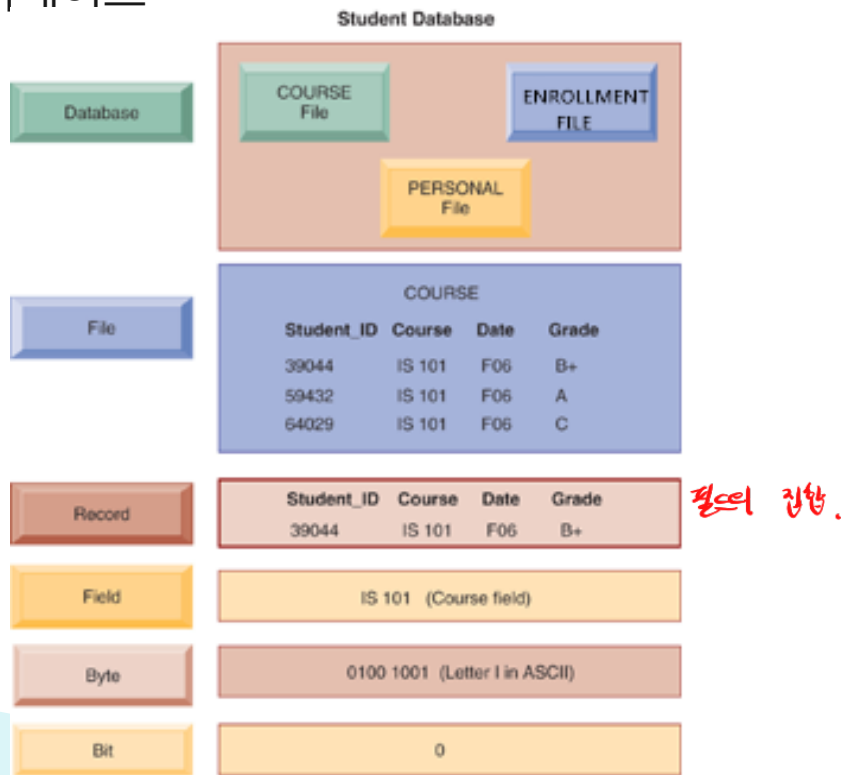
- 논리적 계층

논리적 계층의 최소단위.

- 필드(field): 문자의 집합으로서, 의미있는 단어 또는 숫자를 표현한다 (예: 이름, 나이)
- 레코드(record): 연관된 필드의 집합 (예: 특정 학생의 이름, 학번, 수강과목, 성적)
- 파일(file): 같은 유형의 레코드 집합
- 데이터베이스(database): 연관성 있는 파일들의 집합

데이터의 계층구조

- 예) 학사관리 데이터베이스



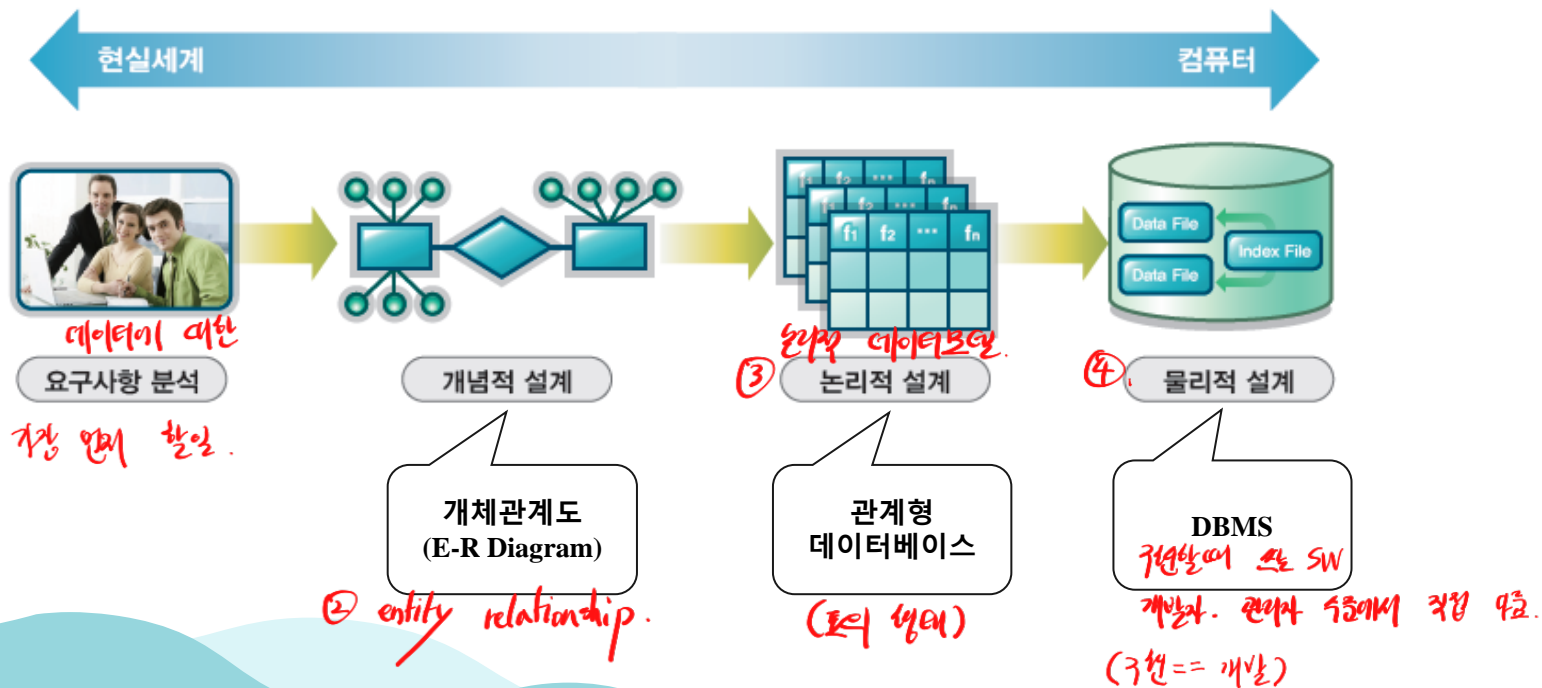
데이터의 계층구조

- 예) 자동차 부품 주문 데이터베이스

계층구조 요소	예시	내용																								
데이터베이스	<div>공급자 파일 부품 파일 주문 파일</div>	부품 데이터베이스																								
파일	<table><tr><th>No.</th><th>Description</th><th>Model</th><th>Vendor</th><th>Unit Pr.</th><th>Qty</th></tr><tr><td>100</td><td>브레이크패드</td><td>Sonata</td><td>DY</td><td>10,000</td><td>5</td></tr><tr><td>105</td><td>스파크플러그</td><td>Avante</td><td>DY</td><td>3,000</td><td>10</td></tr><tr><td>110</td><td>헤드라이트</td><td>Avante</td><td>JS</td><td>65,000</td><td>3</td></tr></table>	No.	Description	Model	Vendor	Unit Pr.	Qty	100	브레이크패드	Sonata	DY	10,000	5	105	스파크플러그	Avante	DY	3,000	10	110	헤드라이트	Avante	JS	65,000	3	부품 파일
No.	Description	Model	Vendor	Unit Pr.	Qty																					
100	브레이크패드	Sonata	DY	10,000	5																					
105	스파크플러그	Avante	DY	3,000	10																					
110	헤드라이트	Avante	JS	65,000	3																					
레코드	<table><tr><th>No.</th><th>Description</th><th>Model</th><th>Vendor</th><th>Unit Pr.</th><th>Qty</th></tr><tr><td>100</td><td>브레이크패드</td><td>Sonata</td><td>DY</td><td>10,000</td><td>5</td></tr></table>	No.	Description	Model	Vendor	Unit Pr.	Qty	100	브레이크패드	Sonata	DY	10,000	5	‘브레이크 패드’ 부품의 레코드												
No.	Description	Model	Vendor	Unit Pr.	Qty																					
100	브레이크패드	Sonata	DY	10,000	5																					
필드	Sonata	‘Model’ 필드의 값																								
바이트	10101000	‘Sonata’의 ‘a’자 바이트																								
비트	0	‘a’자 바이트의 최하위 비트																								

데이터 모델링

- 데이터베이스 설계의 과정



데이터 모델링

- 데이터 모델(Data Model)
 - 데이터들 사이의 논리적인 연관성을 표현하는 방법
 - 현실세계의 업무 프로세스를 데이터베이스로 구현하기 위한 분석 방법
 - 데이터베이스 관리자, 응용 프로그래머, 그리고 **현업**(일반 사용자) 사이의 상호작용과 의사전달 수단으로 사용됨
 - 종류: 개념적 모델, 논리적 모델 **둘다 필요함**
 - 1단계**
(데이터그라프)
 - 2단계**
(포)

데이터 모델링

- 개념적 데이터 모델
 - 사용자 관점에서 현실세계의 업무를 설명하기 위해 개념적으로 데이터 집합을 정의하고 관계를 설정한 모델
 - 개체관계도(entity-relationship diagram: ERD)
 - 개념적 데이터 모델을 정의하는 방법
 - 현실 세계를 개체, 속성, 관계라는 개념으로 표현하는 방법

개념적 데이터 모델

- 개체와 속성

- 개체(Entity; Entity)

- 데이터로서 표현하고자 하는 현실세계의 대상
 - 사람, 장소, 물건, 사건, 개념 등의 물리적/추상적 대상
 - 예) 직원, 고객, 공급사, 주문

- 속성(Attribute)

- 개체의 특성 요소 (예: 이름, 주민번호, 전화, 주소)

개념적 데이터 모델

- 개체와 속성

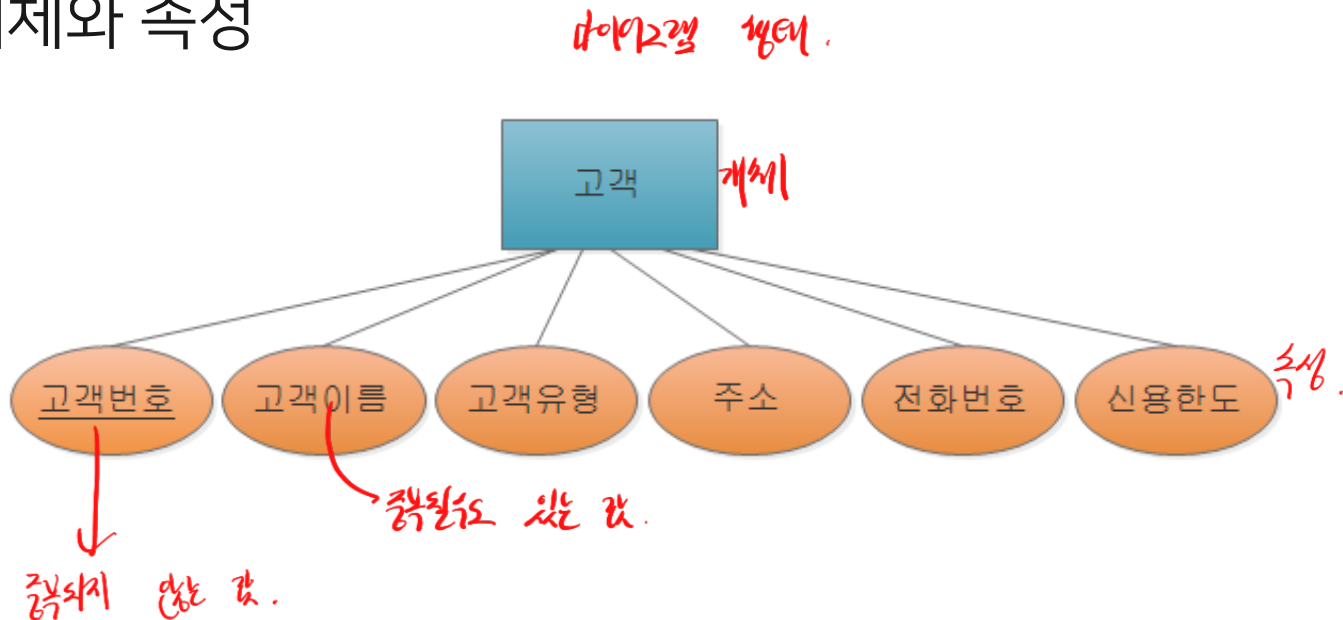
개체	대호 속성. 개체. 속성
고객	<u>고객번호</u> , 이름, 주소, 전화번호
직원	<u>직원번호</u> , 주민등록번호, 이름, 전화번호, 소속부서
제품	<u>제품번호</u> , 제조일자, 모델명, 색상

ex) 주원

주원번호 · 주원자 · 고객번호 · 직원번호

개념적 데이터 모델

- 개체와 속성



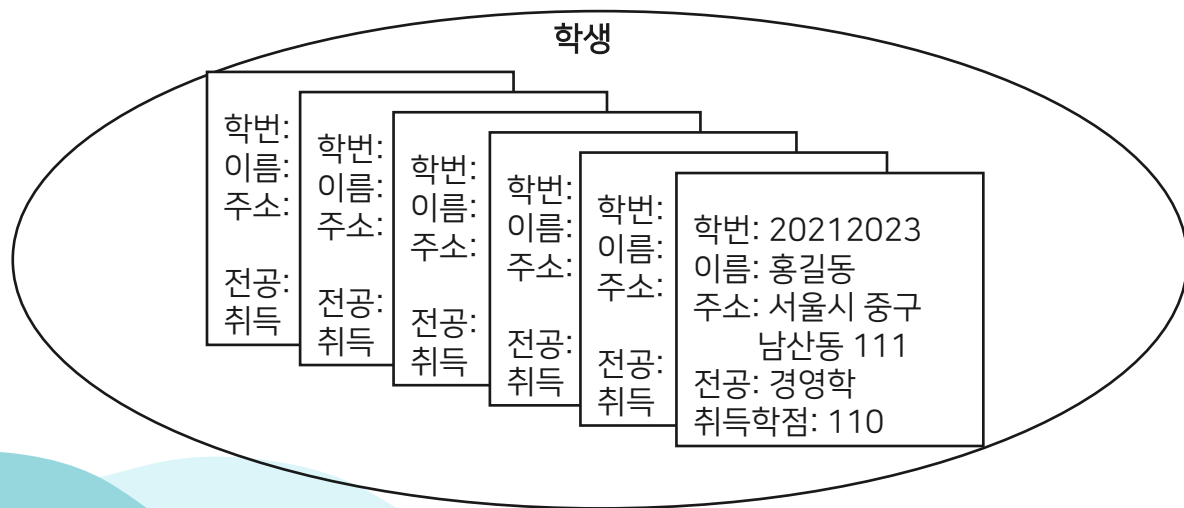
개념적 데이터 모델

- 키(Key) = 키필드.
 - 개체에서 개별 인스턴스(사례)들을 식별하는 속성
 - 예: 고객번호, 주소, 전화번호 등
 - 데이터의 검색, 수정, 정렬을 위한 기준으로 사용된다
 - 기본키(Primary Key): 한 개체 집합에는 복수의 key가 존재할 수 있으며, 이런 경우 하나만 선택하여 식별 속성으로 사용 개체에 해만 있음.
 - 예) 고객번호

개념적 데이터 모델

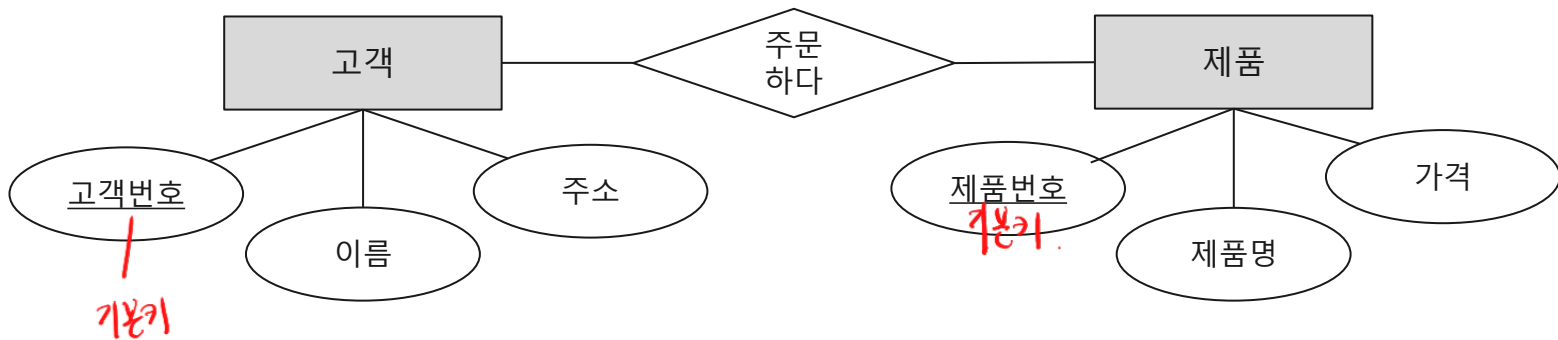
≡ 사례. (=레코드).

- 개체(Entity)와 인스턴스(Instance)
 - 개체는 인스턴스라 불리는 같은 유형의 개별적인 객체들의 집합
 - 예) 학생 개체는 같은 유형의 여러 학생들의 집합



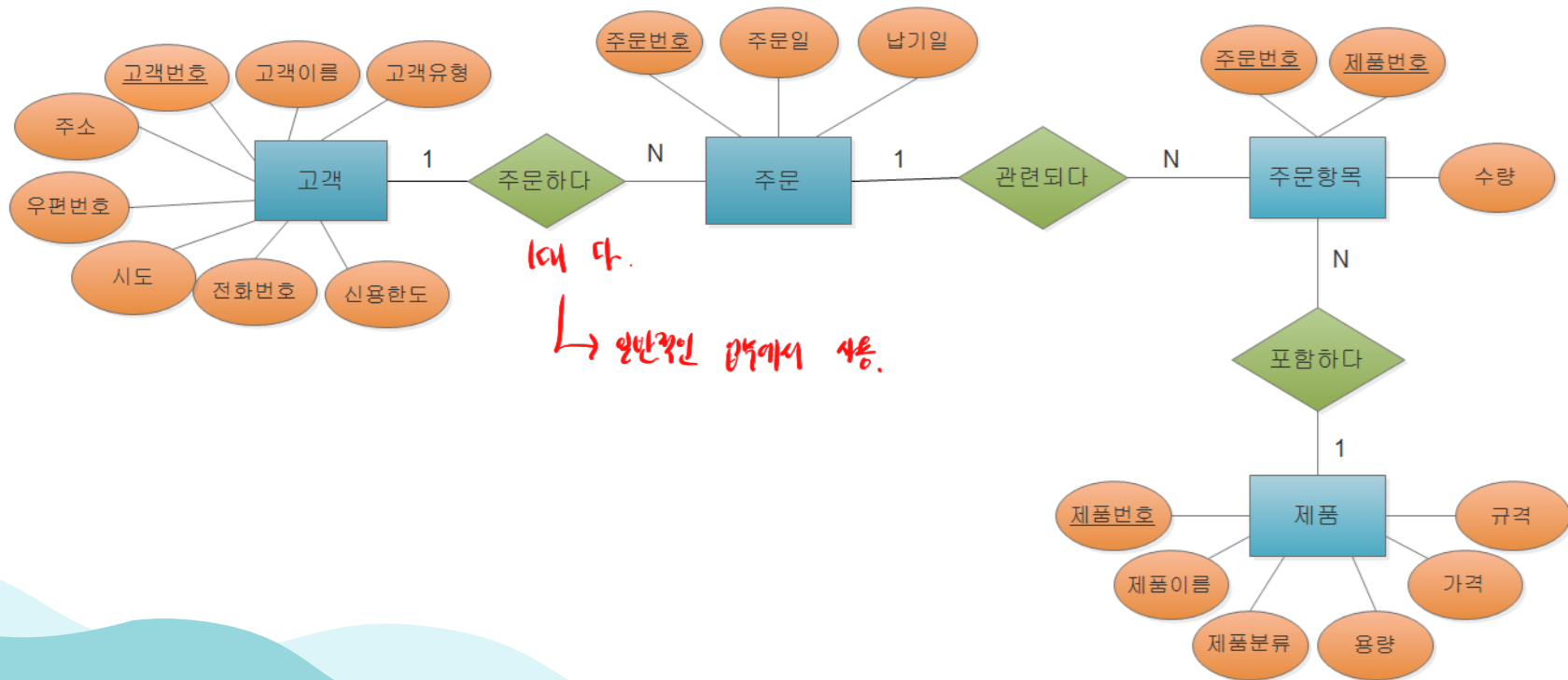
개념적 데이터 모델

- 관계(relationship) ERD.
 - 데이터 개체 간의 업무적 연관성



개념적 데이터 모델

- 주문관리 데이터베이스 개체관계도(ERD)



논리적 데이터 모델

- 관계형 데이터베이스(relational database) **RDB**.

- 오늘날 가장 많이 사용되는 논리적 데이터모델 설계 방식
- 데이터를 행과 열로 구성된 2차원의 테이블 형태로 표현
- 데이터베이스 내 개체들 간의 관계를 구조적으로 정의

개체 = 테이블
속성 = 열 = 필드
인스턴스 = 행 (레코드)

열(=속성, 필드)

기본키

행(=레코드)

고객번호	고객이름	고객유형	주소	전화번호	신용한도	담당직원
1001	무지개약국	약국	노원구 공릉동 570	948-5828	12,000,000	3006
1002	언덕약국	약국	수성구 지산동 456	349-1629	8,000,000	3009
1003	문화약국	약국	송파구 가락동 560	910-4411	15,000,000	3007
1004	미라약국	약국	광산구 고룡동 34-7	904-7840	8,000,000	3003
1005	밀양약국	약국	사하구 감천동 5	971-0555	17,000,000	3008
1006	민생의원	의원	마포구 도화동 208	769-2357	6,500,000	3005
1007	부광약국	약국	성동구 광장동 660	948-6677	9,000,000	3008

논리적 데이터 모델

- 관계형 데이터베이스의 구성 요소
 - 테이블: 개체를 행과 열의 2차원 표로 표현
 - 열(column): 개체의 속성(= 필드, 속성)
 - 행(row): 각 객체(인스턴스)의 데이터 값 집합(=레코드)
 - 키(key)
 - 기본 키(primary key): 테이블의 각 행(레코드)을 식별
 - 외래 키(foreign key): 다른 테이블의 기본 키로서, 특정 레코드에 대해 다른 테이블의 레코드를 연결시키는 역할 데이터를 조종하는 데 사용.

논리적 데이터 모델

- 외래키를 이용한 개체간 관계 구조 정의

기본키



외래키 컬 존재.



(값X : 2개는 동일 한 번만 컬)
값0 : 2개 이상 가능.

주문번호 ▾	주문일 ▾	납기일 ▾	고객번호 ▾
1	2017-03-01	2017-04-07	1004
2	2017-04-02	2017-04-20	1009
3	2017-04-11	2017-04-21	1013
4	2017-04-11	2017-05-07	1012
5	2017-04-19	2017-05-03	1008
6	2017-04-23	2017-05-07	1004
7	2017-04-29	2017-05-17	1006
8	2017-05-01	2017-06-07	1015
9	2017-05-11	2017-06-13	1004
10	2017-05-11	2017-06-07	1001

관계형 데이터베이스

- 예) 자동차부품 주문 데이터베이스

부품 **가** **키** **의** **가** **키**

부품 번호	부품 이름	단가	공급자 번호
137	Door latch	22.00	8259
145	Side mirror	12.00	8444
150	Door molding	6.00	8263
152	Door lock	31.00	8259
155	Compressor	54.00	8261
178	Door handle	10.00	8259

주문 **가** **키** **의** **가** **키**

주문 번호	주문 날짜	부품번호	수량
3502	2019/01/15	137	10
3503	2019/01/16	152	20
3504	2019/01/17	178	5

공급자 **가** **키**

공급자 번호	공급자 이름	도로명	도시	주	우편번호
8259	CBM Inc.	74 5 th Avenue	Dayton	OH	45220
8261	B.R. Molds	1277 Gandolly Street	Cleveland	OH	49345
8263	Jackson Components	8233 Micklin Street	Lexington	KY	56723
8444	Bryant Corporation	4351 Mill Drive	Rochester	NY	11344

관계형 데이터베이스

- 예) 음악 데이터베이스

Songs : Album
(다대 1)

기본 Songs

Song ID	Song Title	Length	Album ID
1	I Won't	3:45	1
2	Begin Again	4:14	1
3	You Got Me	4:00	1
4	Fallin For You	3:35	1
5	I Gotta Feelin	4:49	2
6	Imma Be	4:17	2
7	Boom Boom	4:11	2
8	Meet Me	4:44	2

기본 Album

Album ID	Album Title	Musician ID	Category ID
1	Breakthrough	1	1
2	The E.N.D	2	1
3	Monkey Business	2	1
4	Elephunk	2	1
5	The Fame Monster	3	1
6	Raymond v. Raymond	4	2

기본 Musicians

Musician ID	Musician Name	Musician Photo
1	Colby Caillat	Colby.jpg
2	Black Eyed Peas	BYP.bmp
3	Lady Gaga	Gaga.tiff
4	Usher	Usher.bmp

기본 Categories : Album

Category ID	Category Name
1	Pop
2	R&B
3	Rock
4	Country
5	Blues
6	Classical

Musicians : Album
→ 1대 다.
(여기 앨범이
Musician이
들어갈 수 있는가?)

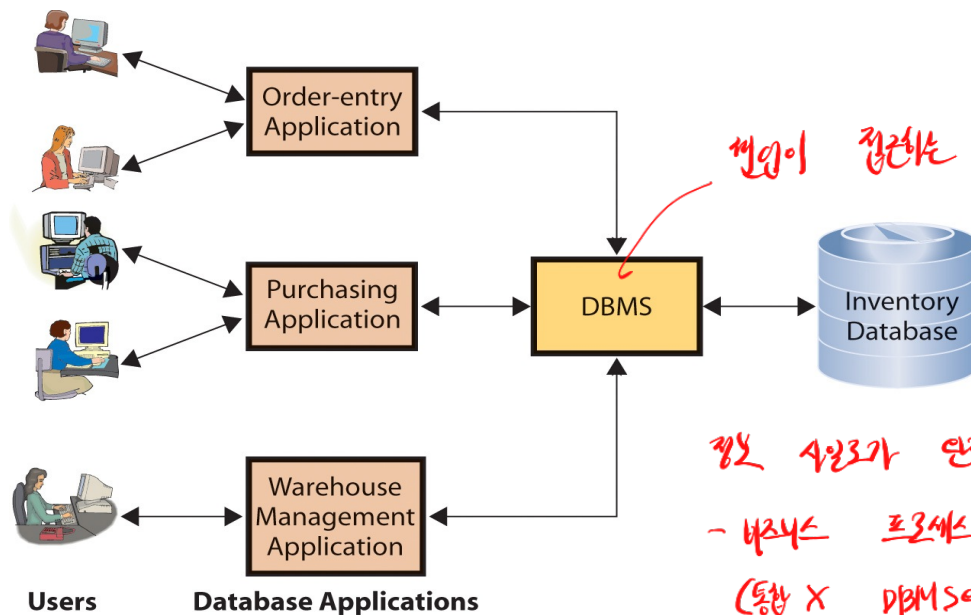


데이터베이스 기술의 활용

- 데이터베이스(DB; database)
 - 중앙 집중식으로 관리되는 구조화된 데이터 집합
 - 사용자들은 다양한 응용 프로그램을 이용하여 데이터베이스에 접근
- 데이터베이스 관리시스템(DBMS; database management system)
 - 응용 프로그램 사용자의 데이터베이스 접근을 통제하는 프로그램 (소프트웨어)
 - 데이터베이스의 생성, 업데이트, 삭제 등의 관리 지원
 - 응용 프로그램과 물리적 데이터를 연결
 - 대표적인 DBMS 제품: IBM의 DB2, 마이크로소프트의 액세스와 SQL서버, 오라클, MySQL

데이터베이스의 기술의 활용

- 기업 정보시스템의 데이터베이스 기술 구조



정보 시스템과 완전히 해결되었나?

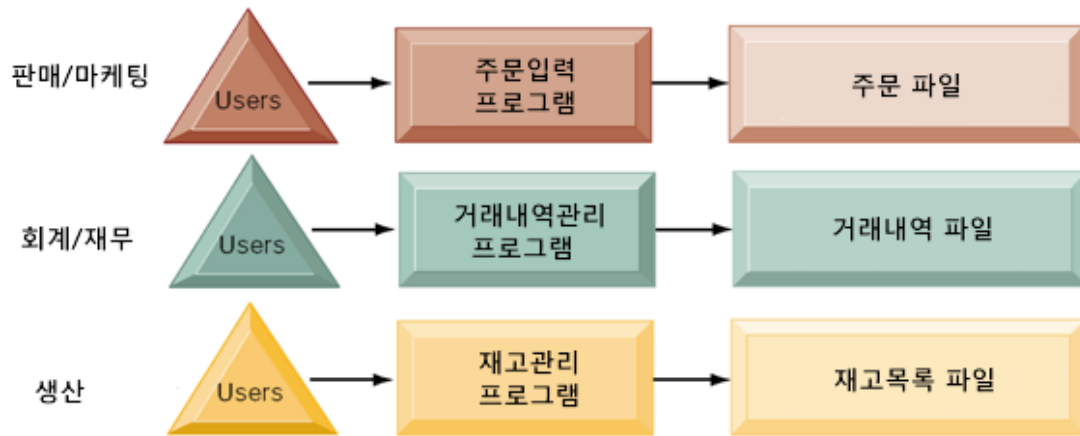
- 비즈니스 프로세스: 분리되어 실행됨
(통합 X, DBMS와 DB만으로 해결 X)

→ ERP 등장. (비즈니스 프로세스 통합)

DBMS의 필요성

- 전통적 데이터 관리 방식의 문제점

- 과거 정보시스템의 개발은 각 부서마다 독립적으로 이루어져, 각 부서에서 사용하는 응용 시스템마다 필요한 데이터를 별도의 개별적인 파일 형태로 관리 → 정보 사일로.



기전 내에는 DBMS 관라자가 따로 존재.

DBMS의 필요성

→ 현재 데이터베이스가 반드시 필요. DBMS 없는 DB는 존재 X

- DBMS의 역할

- 중복 데이터 관리: DBMS가 데이터의 생성, 삭제 권한을 통제함으로써 중복 데이터 생성 차단 *일관된 데이터 관리.*
- 데이터 동시 사용 관리: 데이터에 대한 동시적 접근을 DBMS가 관리함으로써 데이터의 일관성 확보
- 데이터 추출: 응용 프로그램이 요청하는 데이터를 DBMS가 바로 추출하여 제공
→ SQL (데이터 추출을 위한 프로그래밍 언어)
- 보안 통제: DBMS가 응용 프로그램의 데이터베이스 접속을 통제/차단함으로써 보안 위험 감소
보안성.

Case Review

BAE Systems 제조 데이터 통합 관리를 위한 전사적 시스템 도입

- 유럽 최대의 상용 항공기 제조기업인 BAE는 세계 40개국에 판매, 제조, 고객지원을 위한 8만여 명의 직원을 고용
- 기존의 레거시(legacy) 정보시스템은 설계와 제조현장이 분산되어 항공기 조립을 위한 복잡한 부품 명세서 데이터를 전체 생산라인에 걸쳐 정확하게 운영하고 저장하기 어려웠음
- 데이터 불일치 문제를 해결하기 위해 전사적 지식경영 시스템을 도입하고, 기존의 레거시 시스템에 분산 보관된 데이터를 통합하여 단일 저장소에 저장
- 전세계 설계 및 제조 엔지니어가 동시에 단일 데이터베이스에 접속하여 항공기 설계 데이터 활용
- 레거시 시스템의 데이터 파일 수를 줄임으로써 데이터 관리 및 저장 비용을 절감하였고, 항공기 부품 데이터를 정확하게 정의함으로써 품질향상에 기여

