

1주차 : 데이터베이스 기본 개념

1. 데이터베이스 필요성
2. 데이터베이스 정의와 특징
3. 데이터 유형

2주차 : 데이터베이스 관리 시스템

1. DBMS 개념
2. DBMS 장단점
3. DBMS 발전과정

3주차 : 데이터베이스 시스템

1. 데이터베이스 시스템 구성
2. 스키마
3. 데이터베이스 사용자
4. 데이터 언어

4주차 : 데이터베이스 모델링

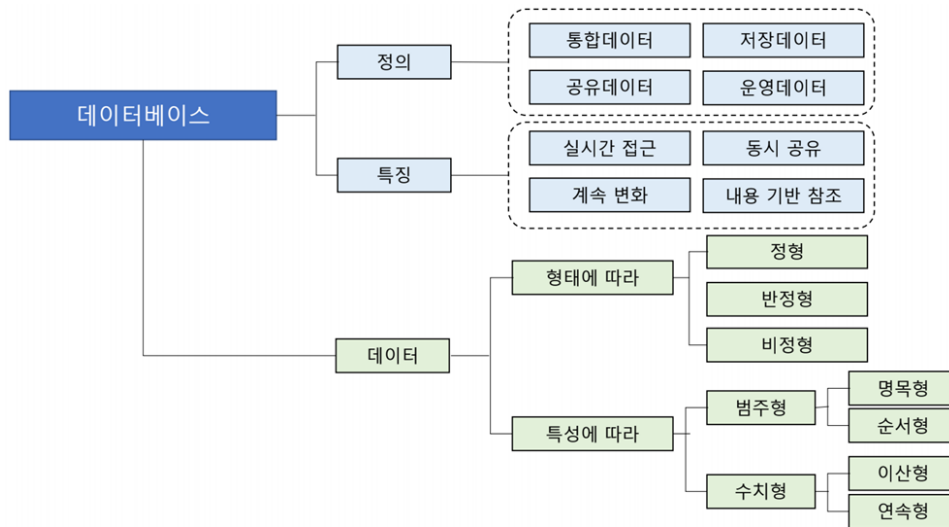
1. 데이터 모델링
2. 개체-관계 모델
3. 논리적 데이터 모델

5주차 : 데이터베이스 모델링

1. 관계 데이터 모델의 개념
2. 관계 데이터 모델의 제약조건

<1주차 : 데이터베이스 기본 개념>

0. 개요



1. 데이터베이스 필요성

* 홈페이지마다 비밀번호를 전부 다르게 설정(or 몇 개월마다 교체)하는 것이 원칙. But, 기억을 못 해서, 어딘가에 적어놓거나 벽에 붙여놓음. 즉, 머리로 기억하기에는 한계가 있음. 하지만 비밀번호가 생각이 안나면 그 사이트에 접근이 안됨.

나 혼자만으로도 이런 혼란이 있음 -> 기업체는 더 많고 다양한 종류의 데이터가 존재. 그러므로 전문적으로 관리하는 "관리자"가 필요함.

* "카메라 회사가 없어지고 디지털 시대가 올 것이다"라는 것을 예측하지 못함 -> 빠른 변화 및 대응을 못함 or 디카, 핸드폰 분야에 더 투자함 -> 즉, 올바르게 가치있는 정보의 획득을 통한, 올바른 의사결정이 회사의 성장에 중요한 영향을 미침.

데이터와 정보

(1) 데이터(Data)

- 사실, 신호, 기호 등으로 표현되는 원시적인 값 또는 표현
- 자체로는 큰 의미 없음
- <예시> 특정 학생의 시험 점수 리스트 (85, 90, 78)
- * 데이터가 정보로 탄생하려면 필터를 거쳐야 함.

(2) 정보(Information)

- 정보는 데이터를 처리, 해석 또는 조직하여 어떠한 문맥에서 유용하게 만들어진 결과물.
- 정보는 의사결정이나 인식의 토대로 사용될 수 있음.
- <예시> 특정 학생의 시험 평균 점수는 84.3점 -> 데이터를 바탕으로 얻어진 정보.
- * 데이터(학생들의 점수)를 이용해서 정보(학생들의 점수 평균)을 얻음.

[데이터는 원시적이고 단순한 사실이나 값들의 집합이며, 정보는 이러한 데이터를 의미 있게 변

환한 것. 정보는 데이터보다 높은 수준의 조작, 해석 또는 의미를 갖음.]

(1) 기상 관측

- 데이터 : 8월 18일, 기온: 30°C, 습도: 80%, 강수량: 20mm
- 정보 : 8월 18일은 덥고 습한 날씨에 소나기가 내렸다.

(2) 슈퍼마켓 판매 기록

- 데이터 : 사과 50개 판매, 바나나 30개 판매, 포도 20개 판매
- 정보 : 오늘 가장 많이 판매된 과일은 사과이다.

(3) 학교 성적

- 데이터 : 학생 A: 수학 80점, 영어 70점, 과학 90점
- 정보 : 학생 A는 과학에서 가장 높은 점수를 받았고, 영어에서는 가장 낮은 점수를 받았다.

(4) 조사 결과

- 데이터 : 응답자 100명 중 40명이 '예'라고 답했고, 60명이 '아니오'라고 답했다.
- 정보 : 조사 결과, 응답자의 60%가 '아니오'라고 응답했다.

(5) 의료 기록

- 데이터 : 환자의 혈압: 130/80 mmHg, 체온: 37.5°C, 심박수: 75bpm
- 정보 : 환자의 혈압과 체온은 정상 범위에 있으며, 심박수도 정상이다.

[데이터는 단순한 수치나 사실을 나타내는 반면, 정보는 그러한 데이터를 해석하거나 조작하여 얻어진 결과물]

정보처리

(1) 정보처리(Information Processing)는 정보나 데이터를 수집, 저장, 조작, 변환 및 전송하는 과정이나 활동

(2) 인간의 두뇌 뿐만 아니라 컴퓨터나 다른 정보 시스템에서도 발생

(3) 주요 단계



(4) 인간과 컴퓨터

- 인간의 두뇌 : 감각기관을 통해 외부에서 정보를 수집(입력) → 뇌에서 이를 처리 → 반응이나 결정을 내려 출력 → 필요한 정보는 기억으로 저장
- 컴퓨터 시스템 : 키보드, 마우스 등의 장치를 통해 데이터를 입력 → CPU에서 이를 처리 → 모니터나 프린터를 통해 출력 → 하드 드라이브나 SSD 등의 저장소에 정보를 저장

(1) 일상생활에서의 정보처리

- 운전 중에 다가오는 신호등의 색을 보고, 그 색에 따라 멈추거나 계속 운전할지 결정하는 것도 정보처리의 일종.

(2) 비즈니스에서의 정보처리

- 고객 서비스 팀이 고객의 피드백을 분석하여 제품 개선에 반영하는 활동.

(3) 컴퓨터 프로그래밍에서의 정보처리

- 웹 크롤러가 인터넷의 페이지를 수집하고, 해당 정보를 분석하여 데이터베이스에 저장하는 과정.

(4) 의료 분야에서의 정보처리

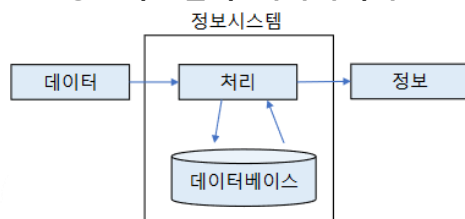
- 환자의 증상과 병력을 기반으로 적절한 치료법을 결정하는 활동.

(5) 과학 연구에서의 정보처리

- 생물학적 실험에서 얻은 데이터를 처리하여 연구 결과와 결론을 도출하는 활동.

[정보처리는 우리 주변에서 꾸준히 일어나고 있으며 다양한 분야와 상황에서 중요한 역할을 하고 있다.]

정보시스템과 데이터베이스



(1) 정보시스템

- 조직의 정보 처리 및 관리와 관련된 전반적인 시스템을 의미
- 하드웨어, 소프트웨어, 데이터, 프로시저, 및 사람 등이 포함
- <예시>
 - 회계 정보 시스템: 회사의 재무 관련 활동을 지원하는 시스템
 - 인사 관리 시스템: 직원 정보 및 인사 관련 업무를 처리하는 시스템

(2) 데이터베이스

- 정보시스템 내의 한 부분으로, 구조화된 데이터를 저장하고 관리하는 역할.
- <예시>
 - 고객 관리 데이터베이스: 회사의 고객 정보, 주문 내역, 연락처 등을 저장하는 데이터베이스
 - 도서관 데이터베이스: 도서의 정보, 대출 내역, 회원 정보 등을 저장하는 데이터베이스

2. 데이터베이스 정의와 특징

데이터베이스 정의

: 특정조직의 여러 사용자가 **공유**하여 사용할 수 있도록, **통합**해서 **저장**한 **운영** 데이터의 집합

(1) 통합된 데이터 (Integrated Data)

- 중복의 최소화

(2) 저장된 데이터 (Stored Data)

- 컴퓨터 매체에 보관된 데이터



(3) 운영 데이터 (Operational Data)

- 실시간으로 생성되고 사용되는 데이터
- 소매점의 매출 데이터, 웹사이트의 실시간 사용자 트래픽 데이터, 은행의 거래 데이터 등.

(4) 공유 데이터 (Shared Data)

- 여러 응용 프로그램이나 사용자 그룹 간에 공유되어 사용되는 데이터
- 중앙 데이터베이스에 저장되는 데이터

데이터베이스의 특징

(1) 실시간 접근성(real-time accessibility)

- 사용자의 요구에 실시간으로 질의에 대한 처리 및 응답이 이루어짐
- 온라인 은행, 항공편 예약 시스템, e-커머스 웹사이트, 긴급 응급 서비스

(2) 내용에 의한 참조(contents reference)

- 데이터가 저장된 주소에 의한 참조가 아닌 내용에 의한 참조.
- 이름으로 고객 정보 검색, 특정단어가 포함된 문서검색, 제품코드로 상세 정보 검색

(3) 동시 공유(concurrent sharing)

- 같은 데이터를 여러 사용자가 동시에 사용 가능.
- 온라인 쇼핑몰, 은행계좌관리, 항공예약시스템, 공공기관의 데이터베이스

(4) 계속적인 변화(continuous evolution)

- 새로운 데이터의 삽입, 삭제, 갱신을 통해 현재의 정확한 자료를 유지하면서 변화. 항상 최신의 데이터를 유지.
- 소셜 미디어 플랫폼, 온라인 쇼핑몰의 재고 관리, 은행 계좌 관리, 예약 시스템, 뉴스 웹사이트

3. 데이터 유형

형태에 따른 분류

지역	학과	성명	성적	학점
서울	건축학과	라이언	85	B
서울	물리학과	임혜림	88	B
대전	전자공학	전하림	75	C
서울	사회복지	이동원	88	B

```
data = {  
  'name': 'John',  
  'age': 30,  
  'city': 'New York'  
}
```

아삭한 식감, 달달한 맛을 가진 참외는 여름
참외는 성주, 김천, 칠곡 등 여러곳에서 생
평소에는 생산지를 가리지 않고 동네에서

(1) 정형 데이터 (Structured Data)

- 고정된 구조나 형식을 가진 데이터, 관계형 데이터베이스의 테이블과 같이 열과 행으로 이루어진 데이터
- <예시>고객 정보, 재고 데이터, 매출 내역 등.

(2) 반정형 데이터 (Semi-structured Data)

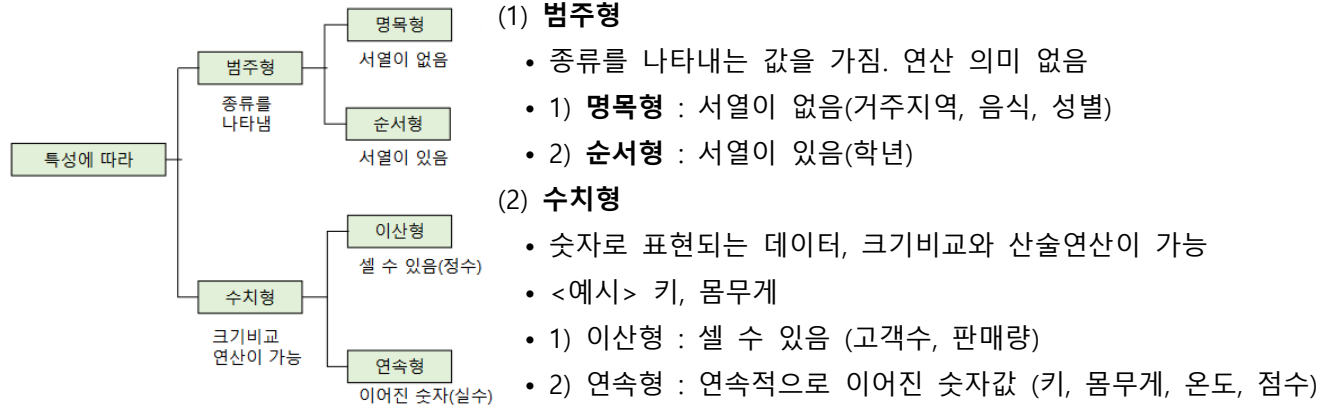
- 구조는 있지만, 관계형 데이터베이스처럼 엄격하게 고정되지 않은 데이터.
- XML, JSON 등의 형식으로 표현되며, 키-값 쌍, 배열, 중첩된 요소 등으로 데이터를 구조화..
- <예시> 웹 페이지의 메타데이터, JSON 형식의 API 응답, 설정 파일 등.

(3) 비정형 데이터 (Unstructured Data)

- 고정된 구조나 형식이 없는 데이터

- 텍스트, 이미지, 동영상, 오디오 파일 등
- <예시> 소셜 미디어 게시물, 사진, 동영상, 음성 녹음 파일, 이메일 내용 등.

특성에 따른 분류



<정리>

- 데이터베이스의 정의에 대해 설명하시오.
- 데이터베이스의 네가지 특징에 대해 설명하시오.
- 데이터를 유형별로 분류하여 설명하시오.

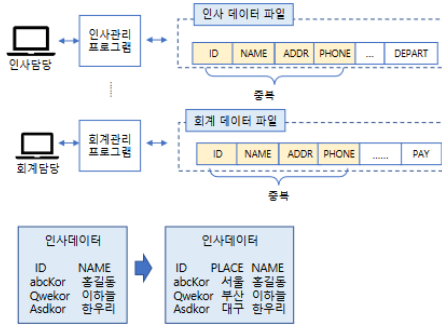
<평가>

- 데이터베이스의 정의로 옳지 않은 것은? [2]
 - ① 조직의 존재 목적이나 유용성 면에서 존재 가치가 확실한 필수적 데이터이다
 - ② 정보 소유 및 응용에 있어 지역적으로 유지되어야 한다.
 - ③ 컴퓨터가 접근할 수 있는 저장 매체에 저장된 자료이다.
 - ④ 동일 데이터의 중복성을 최소화해야 한다.
- 다음 중 이산형 데이터에 속하는 것은? [3]
 - ① 고객 성별
 - ② 가입 기간
 - ③ 가입 고객 수
 - ④ 고객 거주 지역

<2주차 : 데이터베이스 관리 시스템>

1. DBMS 개념

파일 시스템 (File System)



(1) 중복 저장

-> 저장 공간 낭비, 데이터의 일관성, 무결성 유지 어려움

(2) 응용 프로그램이 데이터 파일에 종속적

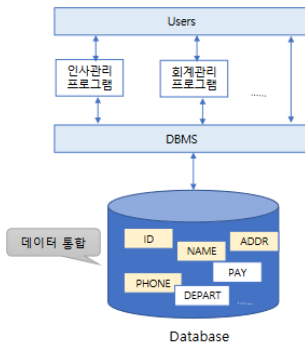
(파일의 구조 변경 -> 응용 프로그램 변경)

(3) 병행 처리의 어려움

Database

: 데이터 및 데이터간 관계의 집합

데이터베이스 관리 시스템 (DataBase Management System)



(1) 사용자와 데이터베이스 사이에서 데이터베이스를 관리해주는 SW.

(2) 데이터를 데이터 베이스에 통합하여 저장하고 관리. (중복의 최소화)

✓ 중복성으로 인한 기존의 문제점

✓ 일관성, 보안성, 경제성, 무결성

(3) DB의 구조를 변경하여도 응용프로그램을 변경할 필요 없다. (독립성)

질의(Query) 예시

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

(1) "신규고용" 정책을 제안한 부서명? (검색)

(2) "총무과"에서 제안한 정책번호를 999로 변경하세요? (수정)

(3) "인사과"에서 제안한 정책명을 삭제하세요. (삭제)

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

(4) "DB"를 수강한 학생 이름은? (검색)

(5) 학번이 300인 학생이 수강한 과목의 학점을 3으로 수정하시오. (수정)

DBMS의 주요기능

(1) 정의 기능(definition faculty)

- 데이터베이스 구조에 대한 정 , 변경, 삭제(CREATE, ALTER, DROP)
- 데이터의 형(Type), 제약조건 명시

(2) 조작기능 (manipulation faculty)

- 사용자와 데이터베이스 간의 인터페이스를 제공.
- 데이터 삽입, 삭제, 수정, 검색(INSERT, DELETE, UPDATE, SELECT)

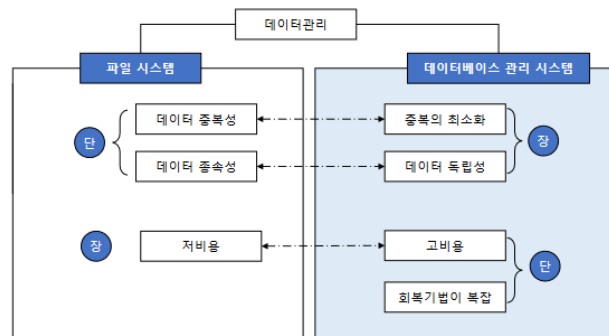
(3) 제어기능 (control faculty)

- 무결성 유지되도록 제어(정확한 데이터 유지)
- 보안유지, 권한검사(허가된 데이터만 접근)
- 병행제어(여러 사용자 동시 접근 → 처리결과와 정확성유지)
- GRANT, REVOKE, COMMIT, ROLLBACK

2. DBMS 장단점

장점

- (1) 데이터 중복 최소화
- (2) 독립성 확보
- (3) 동시 공유
- (4) 보안 향상
- (5) 무결성 유지
- (6) 표준화
- (7) 장애 발생시 회복기능
- (8) 응용프로그램 개발 비용이 줄어듦.



단점

- (1) 데이터베이스 구축 비용 증가
- (2) 백업과 회복 절차가 복잡
- (3) 중앙집중관리

3. DBMS 발전과정

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 데이터 간의 관계를 나타냄
- 1 : N 의 관계, 부모 자식 관계



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



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	홍지민	정보통신	001
102	정민기	정보통신	001
103	무늬영	웹디자인	002
104	정수현	웹디자인	002
105	서유리	사이버안보	003

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)
- <예시> Redis, MongoDB, Apache Cassandra, Neo4j 등.

7. NewSQL DBMS

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



<정리>

- 파일시스템의 문제점과 데이터베이스 관리 시스템의 장점에 대해 설명하세요.

<평가>

1. 다음에서 설명하는 파일 시스템의 문제점은? [3]

“파일의 레코드 크기나 타입이 변경되면, 해당 파일에 액세스 하는 프로그램 코드도 수정되어야 한다.”

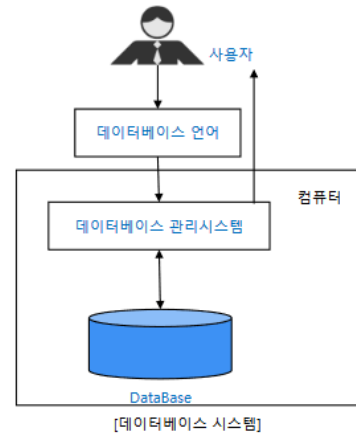
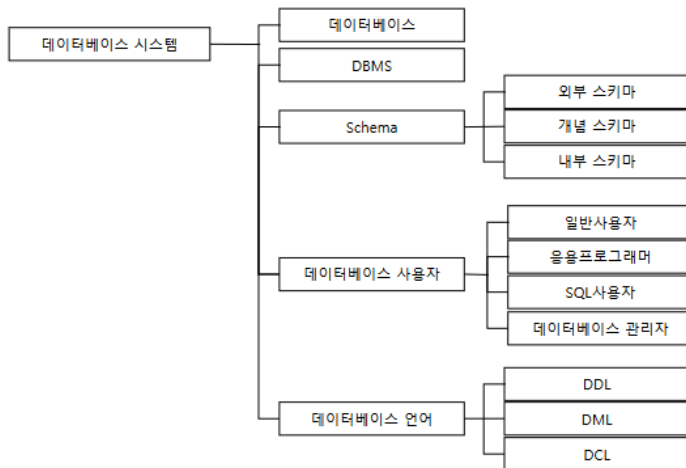
- ① 데이터 중복성
- ② 데이터 독립성
- ③ 데이터 종속성
- ④ 데이터 재사용성

2. 다음 중 데이터베이스 관리 시스템의 장점으로 보기 어려운 것은? [3]

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

<3주차 : 데이터베이스 시스템>

0. 정리



1. 데이터베이스 시스템 구성

데이터베이스 시스템(DBS, DataBase System)

: 데이터베이스에 데이터를 저장, 관리, 조직에 필요한 정보를 생성하는데 필요한 컴퓨터 중심의 시스템

데이터베이스 시스템 구성요소

- (1) 데이터베이스
- (2) 데이터베이스 관리 시스템
- (3) 스키마
- (4) 데이터베이스 사용자
- (5) 데이터베이스 언어

2. 스키마

스키마(schema)

: 스키마는 데이터베이스의 구조(테이블, 뷰, 인덱스 등)와 제약조건을 정의

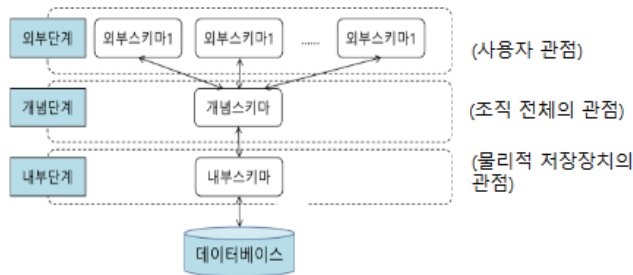


인스턴스 (Instance)

: 특정 시점에 데이터베이스에 실제로 저장되어 있는 데이터

정책			Schema 거의 변동 없음
NUM	SUBJECT	ID	
100	인력양성	40	Instance 자은 변동
200	주택자금	20	
300	친절교육	10	
400	성과금	10	
500	신규고용	20	

외부 스키마, 개념 스키마, 내부 스키마



(1) 외부 스키마(external schema)

- 사용자에게 필요한 데이터베이스를 정의
- 여러 개의 외부 스키마 존재 가능
- 서브 스키마(Sub schema)라고도 함

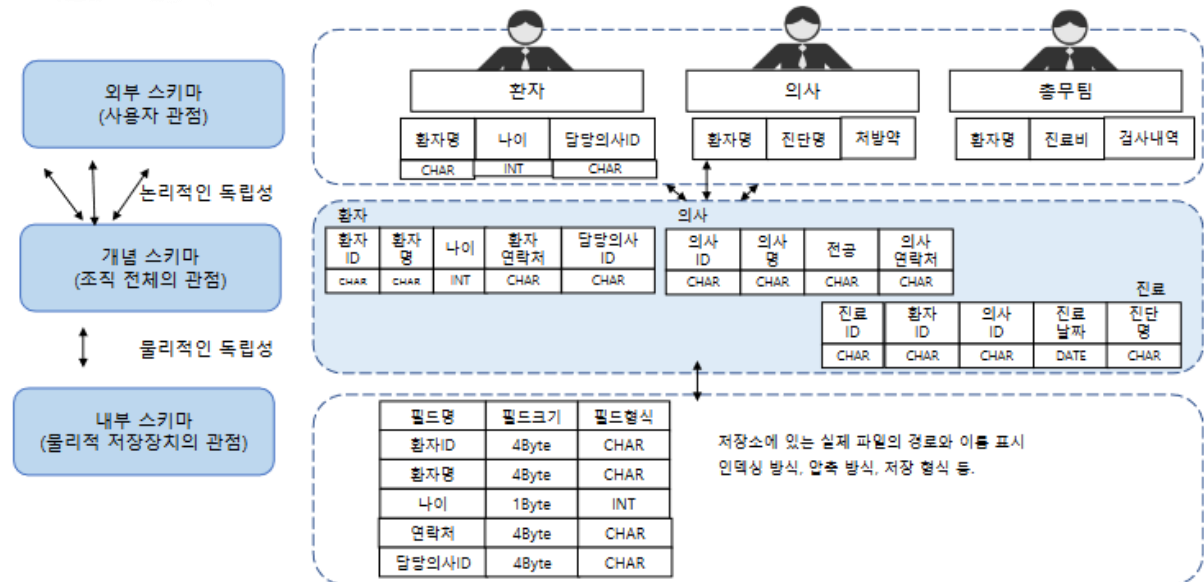
(2) 개념 스키마(conceptual schema)

- 조직 전체의 관점에서 이해하고 표현
- 하나의 데이터베이스에는 하나의 개념스키마가 존재
- 데이터베이스의 접근권한, 보안정책, 무결성 규칙에 관한 명세를 정의

(3) 내부 스키마(internal schema)

- 물리적인 저장장치의 관점에서 표현
- 하나의 데이터베이스에는 하나의 내부스키마가 존재

3단계 데이터베이스 구조



데이터의 독립성과 종속성

(1) 데이터 종속성

- 응용 프로그램에 대한 데이터의 종속성.
- <예시> 파일 시스템
 - 응용 프로그램과 데이터가 상호 의존적.
 - 데이터를 저장한 파일 구조가 변경되면 이에 대응되는 응용 프로그램도 변경되어야 함.

(2) 데이터 독립성

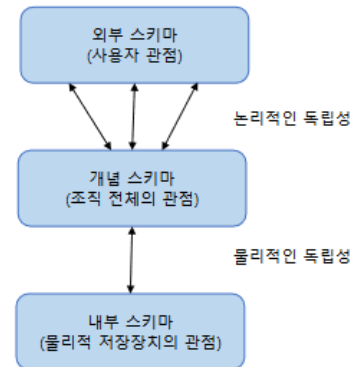
- 데이터 구조가 변경되어도 응용 프로그램이 변경될 필요가 없음.
- 논리적 독립성 + 물리적 독립성으로 실현됨.

(3) 데이터 독립성이 유지되지 않으면?

- 데이터의 중복성 및 복잡도 증가.
- 요구사항 대응 난이도 증가 → 데이터 유지보수 비용 증가.

(4) 데이터 독립성

- 논리적인 독립성
 - 개념스키마 변동 시 외부스키마는 영향을 받지 않음.
 - 논리적 구조가 변경되어도 응용 프로그램에는 영향이 없음.
- 물리적인 독립성
 - 내부스키마 변동 시, 개념 스키마, 외부스키마는 영향을 받지 않음.
 - 저장장치의 구조변경은 응용프로그램과 개념스키마에 영향을 주지 않음.



3. 데이터베이스 사용자

데이터베이스 사용자

(1) 일반사용자

- 응용 프로그래머가 작성한 프로그램을 이용
- 은행 창구, 관공서의 민원 접수처 등에서 데이터를 다루는 업무

(2) 응용 프로그래머

- 일반 사용자가 응용프로그램을 사용할 수 있도록 인터페이스 제공 (C, JAVA...)

(3) SQL 사용자

- SQL를 사용하여 업무를 처리하는 IT부서의 담당자
- 데이터 검색, 데이터 구조 변경, 데이터에 관한 통계처리

(4) 데이터베이스 관리자 (DataBase Administrator, DBA)

- 데이터베이스를 총괄, 데이터 전체 파악 및 관리, 데이터 이동 및 복사 등 제반 업무
- 스키마 정의, 데이터 설계, 구현, 유지보수의 전과정을 담당

4. 데이터 언어

데이터 정의어, 데이터 조작어, 데이터 제어 언어

(1) 데이터 정의어 (DDL : Data Definition Language)

- 데이터 베이스 구조, 데이터 형식, 접근 방식 등 데이터베이스를 구축하거나 수정할 목적으로 사용.
- 스키마를 기술하기 위해 사용되며, 주로 DB 설계자가 사용함
- 생성(CREATE), 변경(ALTER), 삭제(DROP)

(2) 데이터 조작어 (DML : Data Manipulation Language)

- 데이터의 검색(Select), 삽입(Insert), 삭제(Delete), 갱신(Update)에 사용됨
- 사용자와 DBMS 간의 인터페이스를 제공.

(3) 데이터 제어 언어(DCL : Data Control Language)

- 무결성, 보안 및 권한 제어, 회복 등을 하기 위한 언어.
- 데이터 정확성을 위한 무결성 유지, 시스템 장애에 대비한 데이터 회복과 병행수행제어
- 권한부여(GRANT), 권한삭제(REVOKE), 완료(COMMIT), 취소(ROLLBACK)
- cf) TCL(Transaction Control Language)

독립 실행형 / 내장형

(1) 독립 실행형

- SQL 인터페이스를 이용하여 SQL 쿼리를 직접 DBMS에 입력

(2) 내장형

- C, C++, Java 등의 프로그래밍 언어에 내장됨.
- Host language + Data sublanguage 로 구성됨.

절차형 / 비절차형

(1) 절차형

- 프로그램 명령어들을 순차적으로 실행
- C, JAVA, PASCAL, FORTRAN

(2) 비절차형

- 문제 해결의 절차를 명세하지 않고, 원하는 결과만을 명세
- SQL(21세 이상의 사용자의 이름과 나이를 검색)

5. 용어

데이터 사전

- 데이터베이스에 저장되는 데이터에 관한 정보를 저장하는 곳
- DB내의 테이블 이름, 컬럼이름, 데이터 유형, 기본키, 외래키 등의 정보를 포함,
- 스키마, 사상정보, 다양한 제약조건을 저장
- 시스템 카탈로그 라고도 함.

메타데이터

- 데이터의 유형, 관계 수정날짜 등 다양한 정보를 포함
- 데이터의 데이터

<정리>

- 외부 스키마(external schema), 개념 스키마(conceptual schema), 내부 스키마(internal schema)에 대해 설명하세요.

<평가>

1. 사용자가 데이터의 삽입, 삭제, 수정, 검색 등의 처리를 데이터베이스 관리시스템에 요구하기 위해 사용되는 데이터 언어는? [2]

- ① DDL
- ② DML
- ③ DCL
- ④ TCL

2. 개념 스키마에 대한 설명으로 옳지 않은 것은? [4]

- ① 데이터베이스 파일에 저장되는 데이터의 형태를 나타낸 것으로, 단순 스키마라고도 한다.
- ② 범기관적 입장에서 데이터베이스를 정의한 것이다.
- ③ 모든 응용 시스템과 사용자가 필요로 하는 데이터를 통합한 조직 전체의 데이터베이스로 하
나만 존재한다.
- ④ 사용자나 응용 프로그래머가 접근하는 데이터베이스를 정의한 것이다.

<4주차 : 데이터베이스 모델링>

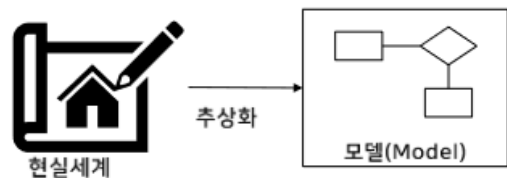
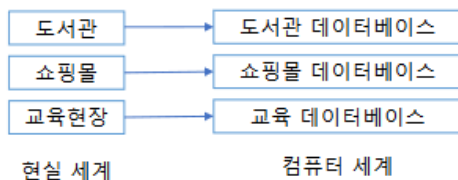
1. 데이터 모델링

데이터 모델링 (data modeling)

(1) 현실세계의 복잡한 정보와 관계를 컴퓨터 세계에서 처리할 수 있도록 변환하는 과정

(2) 추상화 (↔ 구체화)

- 도서관 시스템 : 도서, 회원, 대출, 예약
- 온라인 쇼핑몰 : 상품, 고객, 주문, 리뷰, 재고
- 교육 시스템 : 학생, 강사, 과목, 성적, 시간표



모델링의 3가지 관점

(1) 데이터 관점

- What, 무엇이 필요한지
 - 데이터와 데이터 간 관계, 업무와 데이터 간 관계를 모델링
 - 데이터에 접근하는 방법(How), 사람(Who)과는 무관

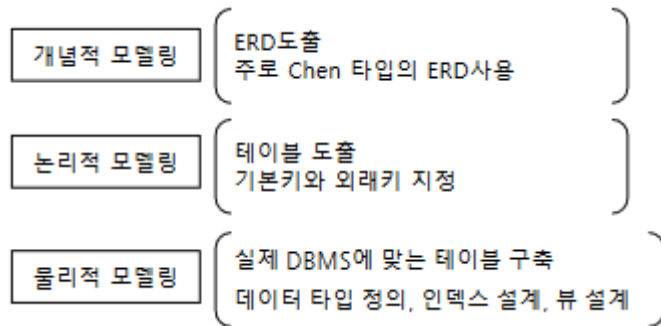
(2) 프로세스 관점(How)

- 업무가 실제로 하고 있는 일 또는 해야할 일을 모델링

(3) 데이터와 프로세스의 상관 관점(Interaction)

- 업무 처리 방법에 따라 데이터가 받는 영향을 모델링

데이터 모델링의 3단계



(1) 개념적 모델링(Conceptual Data Modeling)

- 의사소통을 돕고, 전체적인 데이터 흐름과 구조에 대한 이해를 높이는 목적
- 예) 개체-관계 모델

(2) 논리적 모델링(Logical Data Modeling)

- 개념적 모델을 좀 더 구체화하여, 데이터의 논리적 구조와 관계를 정의
- 데이터의 타입, 기본 키, 외래 키, 인덱스 등을 명세
- 예) 관계 데이터 모델

(3) 물리적 모델링(Physical Data Modeling):

- 논리적 모델을 실제 데이터베이스 구현에 맞게 최적화
- 데이터의 저장, 액세스 방식에 대한 세부 사항을 정리

[예시] 온라인 쇼핑몰

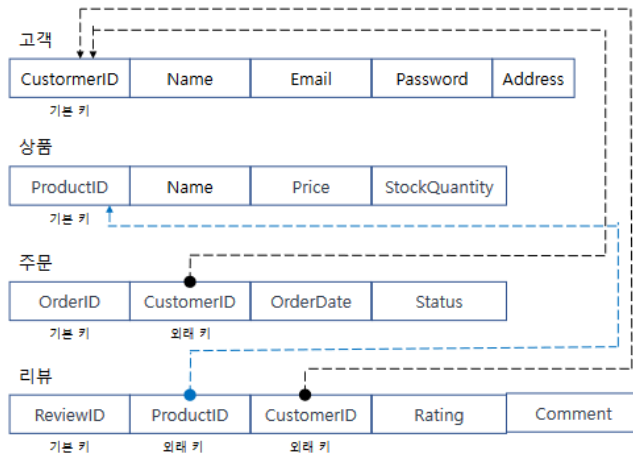
(1) 개념적 모델링 (Conceptual Modeling)

: 온라인 쇼핑몰의 주요 엔터티(Entity)와 그들 간의 관계(Relationship)를 정의

- 주요 엔터티 : 고객, 상품, 주문, 리뷰 등
- 주요 관계
 - 고객은 주문을 할 수 있다.
 - 주문은 여러 상품을 포함할 수 있다.
 - 상품은 여러 리뷰를 가질 수 있다.
 - 리뷰는 한 명의 고객에 의해 작성될 수 있다.
- 다양한 ERD(Entity-Relationship Diagram) 표기법으로 표현
 - Peter Chen 표기법, IE 표기법, Barker 표기법, ...

(2) 논리적 모델링 (Logical Modeling)

: 논리적 모델링에서는 개념적 모델을 좀 더 구체화



- 고객 엔터티 : CustomerID (기본키), Name, Email, Password, Address
- 상품 엔터티 : ProductID (기본키), Name, Price, StockQuantity
- 주문 엔터티 : OrderID (기본키), CustomerID (외래키), OrderDate, Status
- 리뷰 엔터티 : ReviewID (기본키), ProductID (외래키), CustomerID (외래키), Rating, Comment
- 관계와 제약 조건:
 - 한 명의 고객은 여러 주문을 할 수 있음 (고객 1:N 주문)
 - 한 개의 주문은 여러 상품을 포함할 수 있음 (주문 1:N 상품)
 - 하나의 상품은 여러 리뷰를 가질 수 있음 (상품 1:N 리뷰)
 - 하나의 리뷰는 하나의 고객에 의해 작성됨 (고객 1:1 리뷰)

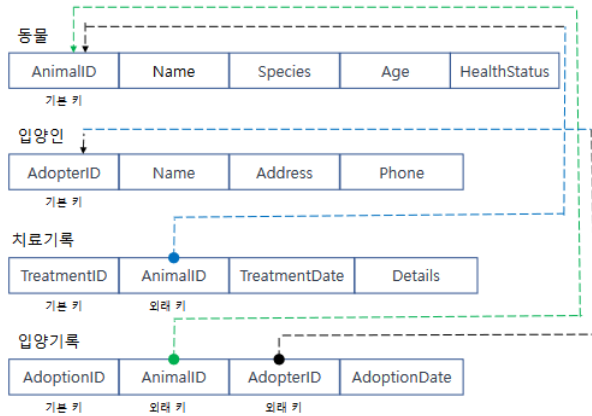
[예시] 동물 보호소

(1) 개념적 모델링 (Conceptual Modeling)

- 동물 보호소의 주요 엔터티(Entity)와 그들 간의 관계(Relationship)
- 주요 엔터티 : 동물, 입양인, 치료 기록, 입양 기록 등
- 주요 관계
 - 동물은 하나의 보호자에 의해 관리될 수 있다.
 - 입양인은 여러 동물을 입양할 수 있다.
 - 동물은 치료 기록을 여러 개 가질 수 있다.
 - 입양 기록은 특정 동물과 입양인에 연결될 수 있다.

(2) 논리적 모델링 (Logical Modeling)

: 개념적 모델을 더 구체적으로 만들어 감



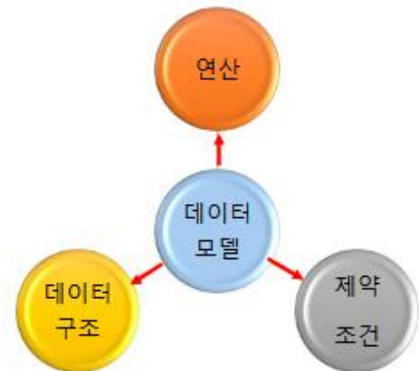
- 동물 엔터티 : AnimalID (기본키), Name, Species, Age, HealthStatus
- 입양인 엔터티 : AdopterID (기본키), Name, Address, Phone
- 치료기록 엔터티 : TreatmentID (기본키), AnimalID (외래키), TreatmentDate, Details
- 입양기록 엔터티 : AdoptionID (기본키), AnimalID (외래키), AdopterID (외래키), AdoptionDate
- 관계와 제약 조건:
 - 한 명의 입양인은 여러 동물을 입양할 수 있다. (입양인 1:N 동물)
 - 한 마리의 동물은 여러 치료 기록을 가질 수 있다. (동물 1:N 치료 기록)
 - 입양 기록은 특정 동물과 입양인에 연결된다. (동물 1:1 입양 기록, 입양인 1:N 입양 기록)

데이터 모델

(1) 데이터 모델링의 결과물을 표현하는 도구

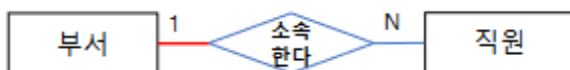
(2) 데이터 모델에 표시할 요소

- 데이터 구조(data structure)
 - 구성 요소
 - 정적 특징
- 연산(operation)
 - 표현된 값들을 처리하는 작업
 - 동적 특징
- 제약조건(constraint)
 - 데이터 무결성 유지 목적

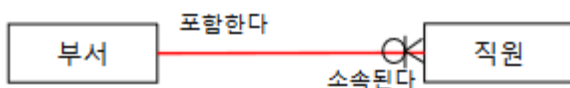


데이터 모델 표시법

(1) Chen



(2) IE / Crow's Foot



(3) 기타

- IDEF1X (Integration Definition for Information Modeling)
- UML (Unified Modeling Language)

2. 개체 - 관계 모델

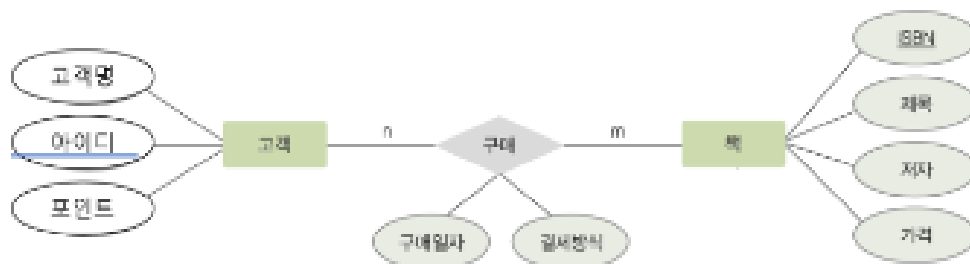
데이터베이스 시스템(DBS, DataBase System)

(1) 개체-관계 모델(ERM, E-R Model, Entity-Relationship model)

- 피터 첸(Peter Chen)이 제안한 개념적 데이터 모델
- 개체와 개체 간의 관계를 이용해 현실 세계를 개념적 구조로 표현
- 핵심 요소 : 개체, 속성, 관계
 - Entity(개체): 데이터 모델에서 대상이나 개념
 - Attribute(속성): 개체의 특성이나 성질
 - Relationship(관계): 개체간의 연관성이나 상호작용

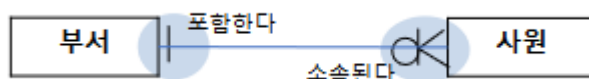
(2) 개체-관계 다이어그램(ERD, E-R diagram, Entity-Relationship diagram)

- 개체-관계 모델을 시각적으로 표현-> 관계자들이 쉽게 이해할 수 있도록 함



ERD 작성 순서

- (1) 엔티티를 그린 후 적절하게 배치
 - 가급적 선이 꼬이지 않게 배치
 - 왼쪽 → 오른쪽, 위 → 아래 순으로 읽어 나가기 편하도록 배치
- (2) 엔티티간 관계 설정
 - 식별자 관계를 우선 설정함
 - 가급적 Cycle 관계도 발생하지 않아야 함
- (3) 관계명 기술 (양 방향)
 - 현재형 사용, 지나치게 포괄적인 단어는 지양
 - 실제 프로젝트에서는 크게 고려하지 않음
- (4) 관계차수와 선택성 표시



2-1. 개체 - 관계 모델 (개체)

엔티티(Entity)의 정의

(1) 변별할 수 있는 사물 - Peter Chen (1976)

- (2) 데이터베이스 내에서 변별 가능한 객체 - C.J Date (1986)
 (3) 정보를 저장할 수 있는 어떤 것 - James Martin (1989)
 (4) 정보가 저장될 수 있는 사람, 장소, 물건, 사건 그리고 개념 등 - Thomas Bruce (1992)



의사



말



사건

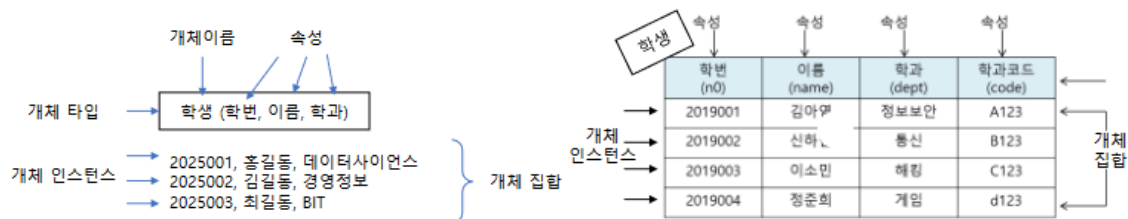


과목

- 업무에 필요한 정보를 저장하고 관리하기 위한 집합적인 것(Thing)

엔터티의 분류

- (1) 유형(Tangible) 엔터티
- 물리적인 형태가 있고 안정적이며 지속적으로 활용됨
 - 교수, 강의실, 학생 등
- (2) 무형 엔터티
- 개념(Conceptual) 엔터티
 - 물리적인 형태는 존재하지 않으나 관리해야 할 개념적 정보
 - 수업, 보험상품 등
 - 사건(Event) 엔터티
 - 업무 수행 과정에서 발생하며 비교적 발생량이 많음 (각종 통계 자료에 이용됨)
 - 수강신청, 주문, 입금 등



- (1) 타입(entity type)
- 같은 종류의 개체들이 공통적으로 가지는 속성을 정의
 - 예. 학생 개체타입은 학번, 이름, 학과 등의 공통속성을 가짐
 - 파일 구조의 레코드 타입(record type)에 대응됨
- (2) 인스턴스(entity instance)
- 개체 타입을 구성하는 속성이 실제로 저장되어 있는 데이터 값
 - 개체 어커런스(entity occurrence)라고도 함
 - 파일 구조의 레코드 인스턴스(record instance)에 대응됨
- (3) 집합(entity set)
- 특정 개체 타입에 대한 개체 인스턴스들을 모아놓은 것

엔터티의 특징

- (1) 고유성 : 각 엔터티는 고유의 속성을 가지고 있어야 하며, 다른 엔터티와 구분할 수 있어야 함.
 (2) 유일한 식별자에 의해 식별이 가능해야 함.
 (3) 영속적으로 존재하는 (둘 이상) 인스턴스의 집합이어야 함.

(4) 업무 프로세스에 의해 이용되어야 함

- 업무 프로세스에 의해 CRUD(Create, Read, Update, Delete)가 발생해야 함
- CRUD가 발생하지 않는다면, 부적절한 엔터티 도출, 또는 업무 누락
- <예시> CRUD Matrix 예시

	Titles	Copies of Books	Check Out Requests	Reservation Requests	Branch Library Transfers	Borrowers	Librarians	Customers	Branch Libraries
Add Books to Library Database	R, C	C		U				R	
Remove Books from Library Database	R	R, D	R, D	R	R, D				
Do Book Search	R	R	R	R	R	R	R	R	
Lend Books	R	R	R, C	R, U	R	R	R		
Return Books	R	R	R, U	R, U		R	R		
Reserve Books	R	R	R	C		R	R	R	
Cancel Book Reservations	R	R		U		R	R		
Manage Book Transfers	R	R	R	R	R, C, U			R	R
DO NOT CARE	R, D	U		D		C, R, D	C, R, D	C, R, D	C, R, D

(1) 반드시 속성을 가져야 함

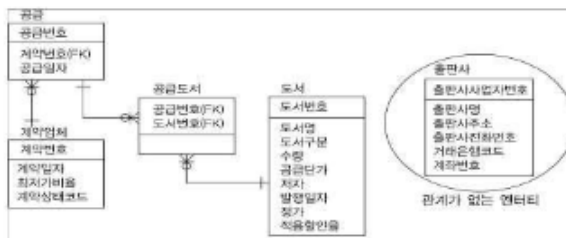
- 속성 없이 엔터티의 이름만 존재할 수 없음

(2) 다른 엔터티와 최소 한 개 이상의 관계를 가져야 함 => 연결되어 있어야 한다.

- 고립 엔터티(Isolated Entity) – 부적절한 엔터티 도출, 또는 관계 누락

(3) 다음의 경우 고립 엔터티를 인정함

- 통계성 엔터티 : (ex) 총거래량
- 코드성 엔터티 : (ex) 에러코드
- 시스템 처리용 내부 엔터티 (트랜잭션 로그 테이블 등)

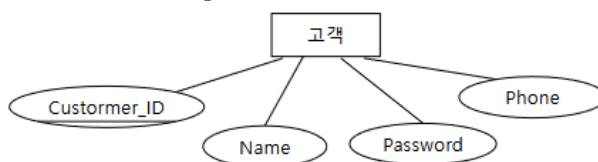


엔터티의 명명

- (1) 엔터티 생성 의미대로, 실제 업무에서 사용하는 용어를 사용한다.
- (2) 약어를 사용하지 않는다.
- (3) 단수 명사를 사용한다.
- (4) 이름이 동일한 엔터티가 중복으로 존재할 수 없다.

2-1. 개체 – 관계 모델 (속성)

[개체에 대한 속성]



[개체에 대한 속성]

속성(attribute)

- (1) 개체나 관계가 가지고 있는 고유한 특성
- (2) 의미 있는 데이터의 가장 작은 논리적 단위

- (3) 파일 구조의 필드(field)와 대응됨
- (4) E-R 다이어그램에서 타원으로 표현하고 타원 안에 이름을 표기

속성의 명명

- (1) 현업에서 사용하는 이름을 부여
- (2) 약어 사용은 가급적 금지
- (3) 서술식 속성명을 피하고 명사형 속성명을 사용
- (4) 수식어와 소유격을 피함
- (5) 속성의 이름은 가급적 전체 모델에서 유일하게 정의

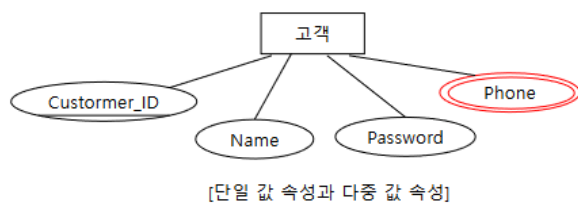
속성 분류

[속성 분류]



단일 값 속성, 다중 값 속성

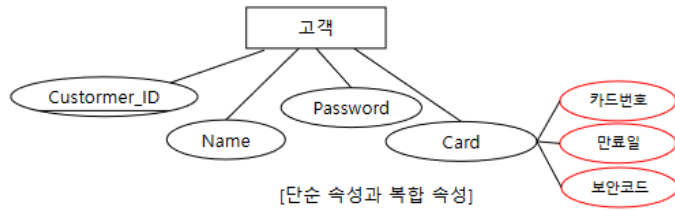
[단일 값 속성과 다중 값 속성]



- (1) 단일 값 속성(single-valued attribute)
 - 값을 하나만 가질 수 있는 속성
 - <예시> 고객 개체의 이름, 적립금 속성
- (2) 다중 값 속성(multi-valued attribute)
 - 값을 여러 개 가질 수 있는 속성
 - <예시>
 - 고객 개체의 Phone 속성
 - 책 개체의 저자 속성
 - E-R 다이어그램에서 이중 타원으로 표현

단순 속성, 복합 속성

[단순 속성과 복합 속성]



(1) 단순 속성(simple attribute)

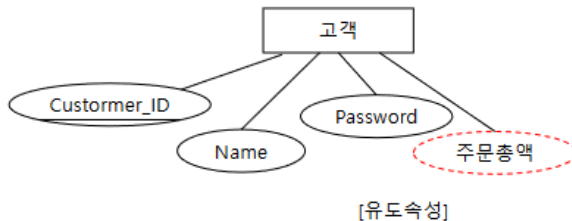
- 의미를 더는 분해할 수 없는 속성
- <예시>
 - 고객 개체의 적립금 속성
 - 책 개체의 이름, ISBN, 가격 속성

(2) 복합 속성(composite attribute)

- 의미를 분해할 수 있는 속성
- <예시>
 - 고객 개체의 카드 속성 : 카드번호, 만료일, 보안코드 등으로 의미를 세분화할 수 있음
 - 고객 개체의 생년월일 속성 : 연, 월, 일로 의미를 세분화할 수 있음

유도 속성, 널 속성, 키 속성, 도메인

(1) 유도 속성 (derived attribute)



- 기존의 다른 속성의 값에서 유도되어 결정되는 속성
- 값이 별도로 저장되지 않음
- <예시>
 - 주문 총액 : 가격과 수량으로 계산
 - 평균 구매금액 : 총 구매금액/총 구매횟수
- E-R 다이어그램에서 점선 타원으로 표현

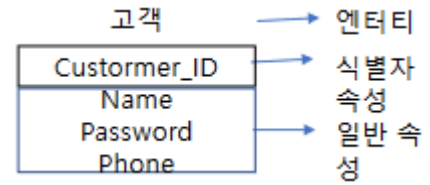
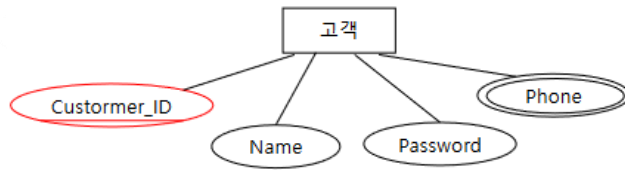
(1) 널 속성 (null attribute)

- 널 값이 허용되는 속성

(2) 널(null) 값

- 아직 결정되지 않았거나 모르는 값 또는 존재하지 않는 값
- 공백이나 0과는 의미가 다름
- <예시> 등급 속성이 널 값 → 등급이 아직 결정되지 않았음을 의미

(3) 키 속성 (key attribute)



- 각 개체 인스턴스를 식별하는 데 사용되는 속성
- 둘 이상의 속성들로 구성되기도 함
- <예시> 고객 개체의 고객아이디 속성
- E-R 다이어그램에서 밑줄로 표현

(1) 도메인 (Domain)

- 각 속성이 가질 수 있는 값의 범위
- <예시>
 - 평점: 0.0 ~ 4.5 사이의 실수
 - 주소: 길이가 20자리 이내인 문자열
- 속성에 대한 데이터 타입과 크기, 그리고 제약사항을 지정하는 개념

2-2. 개체 - 관계 모델 (관계)



관계 (relationship)

- (1) 개체와 개체가 맺고 있는 의미 있는 연관성
- (2) 개체 집합들 사이의 대응 관계, 즉 매핑(mapping)을 의미
- (3) <예시> 고객 개체와 책 개체 간의 구매 관계 -> "고객은 책을 구매한다"
- (4) E-R 다이어그램에서 마름모로 표현

관계의 표기법

(1) 관계명

- 각 관계는 두 방향의 관계명을 가짐
- 명명 규칙
 - 애매한 동사를 피한다 -> <예시> "관계된다", "관련있다" 등을 피함
 - 현재형으로 표현한다 -> <예시> "신청했다", "강의할 것이다" 등을 피함

관계의 유형

(1) 관계에 참여하는 개체 타입 수 기준

- 이항 관계 : 개체 타입 2개가 맺는 관계
- 삼항 관계 : 개체 타입 3개가 맺는 관계
- 순환 관계 : 개체 타입 1개가 자기 자신과 맺는 관계

(2) 처리하는 업무 형태 기준 : 종속, 중복, 재귀, 배타

(a) 종속 관계(Dependent Relationship)

: 두 개체 사이의 주·종 관계를 표현, 식별 관계와 비식별 관계가 있다.

- <예시>

- 진단과 처방(진단 결과에 따라 약의 처방이 이루어짐)
- 원자재 공급과 생산(원자재가 공급된 후에 제품 생산이 이루어짐)

(b) 중복 관계(Redundant Relationship)

: 두 개체 사이에 2번 이상의 종속 관계가 발생하는 것.

- <예시> 사원이 속한 부서와 부서에 속한 사원

(c) 재귀 관계(Recursive Relationship)

: 개체가 자기 자신과 연결되어 있는 경우

- <예시> 직원은 한 명의 관리자가 있다. 관리자도 직원이다.

(d) 배타 관계(Exclusive Relationship)

: 개체의 속성이나 구분자에 의해 개체의 특성을 분할하는 관계

(i) 배타 AND 관계 : 하위 개체 중 하나 이상에 속할 수 있음

- <예시> 사용자가 여러 권한을 가질 수 있음

(ii) 배타 OR 관계 : 하위 개체중 하나에만 속할 수 있음

- <예시> 사원의 고용형태(정규직, 계약직, 파트타임중) 선택

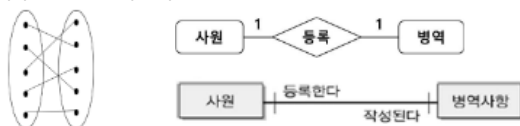
(3) 매핑 카디널리티 기준

- 일대일(1:1) 관계, 일대다(1:n) 관계, 다대다(N:M) 관계

- 매핑 카디널리티(mapping cardinality)

- 관계를 맺는 두 개체 집합에서, 각 개체 인스턴스가 연관성을 맺고 있는 상대 개체 집합의 인스턴스 개수

(a) 일대일(1:1) 관계



- 양쪽에 반드시 1개씩 존재

- <예시> 사람과 여권, 직원과 사원증, 학생과 학번, 차량과 차량번호판

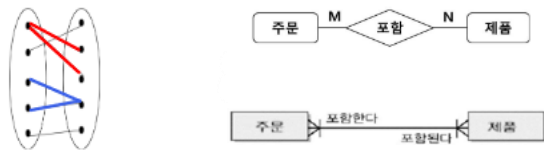
(b) 일대다(1:N) 관계



- 왼쪽에는 반드시 1개, 오른쪽에는 1개 또는 여러 개 존재

- <예시> 교사와 학생, 부모와 자식, 출판사와 책

(c) 다대다(N:M) 관계



- 개체 A의 각 개체 인스턴스가 개체 B의 개체 인스턴스 여러 개와 관계를 맺을 수 있고, 개체 B의 각 개체 인스턴스도 개체 A의 개체 인스턴스 여러 개와 관계를 맺을 수 있음
- <예시> 작가와 책, 배우와 영화

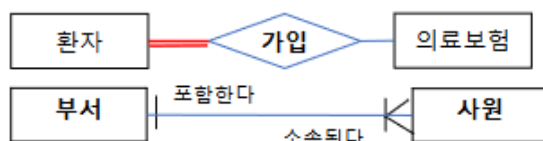
관계의 표현

기호	의미
	필수(Mandatory)
O	선택적(Optional)
<	다중(Multiple)

관계	표현	의미
1:1		양쪽에 반드시 1개씩 존재
1:0 또는 1:1		왼쪽에는 반드시 1개, 오른쪽에는 없거나 1개 존재
1:N		왼쪽에는 반드시 1개, 오른쪽에는 반드시 여러 개 존재
1:1 또는 1:N		왼쪽에는 반드시 1개, 오른쪽에는 1개 또는 여러 개 존재
1:0 또는 1:1 또는 1:N		왼쪽에는 반드시 1개, 오른쪽에는 0개 또는 1개 또는 여러 개 존재

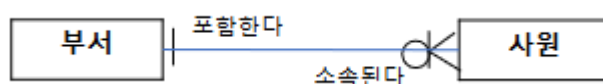
관계의 참여 특성

(1) 필수적 참여(전체 참여, Mandatory Membership)



- 모든 개체 인스턴스가 관계에 반드시 참여해야 되는 것을 의미
- <예시>
 - 모든 학생은 반드시 과목을 선택해야 한다.
 - 모든 작가는 출판사를 소유해야 한다.
 - 모든 환자는 의료보험을 가입해야 한다.
- E-R 다이어그램에서 이중선으로 표현

(2) 선택적 참여(부분 참여, Optional Membership)



- 개체 인스턴스 중 일부만 관계에 참여해도 되는 것을 의미 (선택적으로 참여)
- <예시>
 - 학생은 동아리에 속할 수 있다.
 - 고객은 리뷰를 남길 수 있다.
 - 사용자는 프로필 사진을 설정할 수 있다.

→ 관계의 양쪽이 Optional인 경우, 해당 관계는 잘못 설정되었을 가능성이 큼

관계의 종속성



(1) 약한 개체(weak entity)

→ 다른 개체의 존재 여부에 의존적인 개체

(2) 강한 개체(strong entity)

→ 다른 개체의 존재 여부를 결정하는 개체

(3) 특징

- 강한 개체와 약한 개체는 일반적으로 일대다의 관계
- 약한 개체는 강한 개체와의 관계에 필수적으로 참여
- 약한 개체는 강한 개체의 키를 포함하여 키를 구성

(4) E-R 다이어그램에서 약한 개체는 이중 사각형으로 표현하고, 약한 개체가 강한 개체와 맺는 관계는 이중 마름모로 표현

(5) <예시>

- 도서관책 개체와 대출기록개체 사이의 대출 관계
- 도서관책 개체는 강한 개체, 대출기록 개체는 약한 개체

2-2. 개체 - 관계 모델 (식별자)

식별 관계, 비식별 관계



(1) 식별 관계(Identifying Relationship)

- A 개체의 기본키가 B 개체의 외래키이면서 동시에 기본키가 되는 관계를 말한다.
- ER 도형에서 식별 관계는 실선으로 표시합니다.

(2) 비식별 관계(Non-Identifying Relationship)

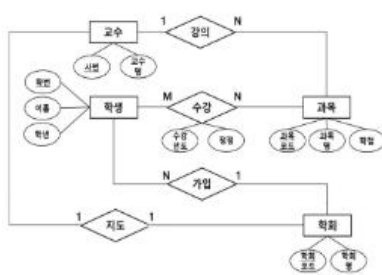
- A 개체의 기본키가 B 개체의 비기본키 영역에서 외래키가 되는 관계를 말합니다.
- ER 도형에서 비식별 관계는 점선으로 표시합니다.

E-R 다이어그램(Entity-Relationship Diagram)

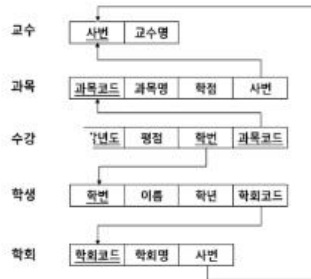
기호	기호이름	의미
	사각형	개체(Entity) 타입
	마름모	관계(Relationship) 타입
	타원	속성(Attribute)
	이중 타원	다중값 속성(복합 속성)
	밑줄 타원	기본키 속성
	복수 타원	복합 속성 예: 성명은 성과 이름으로 구성
	관계	1:1, 1:N, N:M 등의 개체 간 관계에 대한 대응수를 선 위에 기술함
	선, 링크	개체 타입과 속성을 연결

(1) ER Diagram (Conceptual)

- Peter Chen 표기법



(2) Schema Diagram (Logical)



(3) ER Diagram (Conceptual / Logical)

- Information Engineering 표기법
(= Crow's Foot Model)



식별자

(1) 식별자(Identifier)

- 하나의 개체 내에서 각 인스턴스를 유일하게 구분할 수 있는 구분자

주 식별자의 특징

(1) 유일성 (Uniqueness)

- 주식별자에 의해 엔터티 내의 각 인스턴스들을 모두 유일하게 구분할 수 있어야 함
- <예시> ID

(2) 최소성

- 주식별자를 구성하는 속성의 수는 유일성을 만족하는 최소의 수가 되어야 함
- <예시> ID, (ID, 이름)X

(3) 불변성

- 주식별자가 지정되면, 그 식별자의 값은 변하지 않아야 함 예: 탈퇴하기전까지ID는 불변

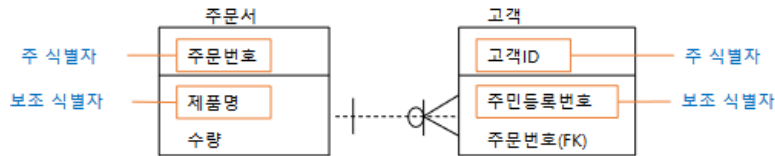
(4) 존재성

- 주식별자의 값은 Null이 될 수 없음 예: ID없는 회원은 있을 수 없음

식별자 분류

(1) 대표성 여부

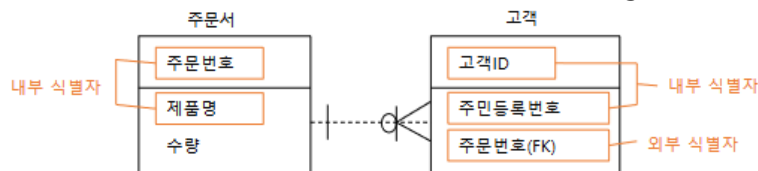
-> 주 식별자(Primary Identifier), 보조 식별자(Alternate Identifier)



- 주 식별자는 개체를 대표하는 유일한 식별자
- 보조 식별자는 주 식별자를 대신하여 개체를 식별할 수 있는 속성
- 하나의 개체에 주 식별자는 한 개만 존재하지만 보조 식별자는 한 개 이상 존재한다.

(2) 스스로 생성여부

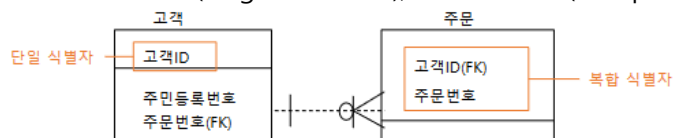
-> 내부 식별자(Internal Identifier), 외부 식별자(Foreign Identifier)



- 내부 식별자 : 개체 내에서 스스로 만들어지는 식별자.
- 외부 식별자 : 다른 개체와의 관계(Relationship)에 의해 외부 개체의 식별자를 가져와 사용하는 식별자
- 외부 식별자는 자신의 개체에서 다른 개체를 찾아가는 연결자 역할을 한다.

(3) 단일 속성 여부

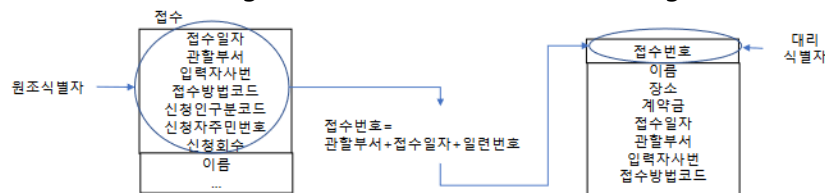
-> 단일 식별자(Single Identifier), 복합 식별자(Composit Identifier)



- 단일 식별자 : 주 식별자가 한 가지 속성으로만 구성된 식별자이다.
- 복합 식별자 : 주 식별자가 두 개 이상의 속성으로 구성된 식별자이다.

(4) 대체 여부

-> 원조 식별자(Original Identifier), 대리 식별자(Surrogate Identifier)



- 원조 식별자 : 업무에 의해 만들어지는 가공되지 않은 원래의 식별자(본질 식별자)
- 대리 식별자 : 주 식별자 속성이 두 개 이상인 경우 속성들을 하나의 속성으로 묶어 사용 (인조 식별자)

(5) 후보 식별자

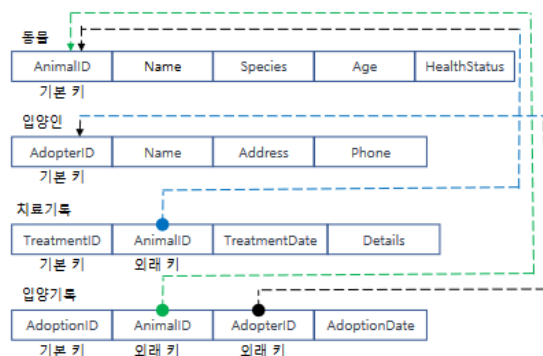
- 후보 식별자는 개체에서 각 인스턴스를 유일하게 식별할 수 있는 속성 또는 속성 집합
- 하나의 개체에는 한 개 이상의 후보 식별자가 있음
- 개체의 대표성을 나타내는 식별자를 주 식별자로, 나머지는 보조 식별자로 지정.
- 후보 식별자 조건
 - 각 인스턴스를 유일하게 식별할 수 있어야 한다.
 - 속성들을 직접 식별할 수 있어야 한다.
 - 널 값(NULL Value)이 될 수 없다.
 - 속성 집합은 후보 식별자로 지정한 경우 개념적으로 유일해야 한다.
 - 후보 식별자의 데이터는 자주 변경되지 않아야 한다.

3. 논리적 데이터 모델

개념

- (1) E-R 다이어그램으로 표현된 개념적 구조를 데이터베이스에 저장할 형태로 표현한 논리적 구조
 - 데이터베이스의 논리적 구조 = 데이터베이스 스키마(schema)
- (2) 종류
 - 관계 데이터 모델,
 - 계층 데이터 모델,
 - 네트워크 데이터 모델 등

관계 데이터 모델



입양인

AdopterID	Name	Address	Phone
101	홍지만	서울	010-777-7777
102	정민기	부산	010-777-7778
103	무뇌영	대구	010-777-7779
104	정수현	대전	010-777-7767
105	서유리	광주	010-777-7757

- (1) 일반적으로 많이 사용되는 논리적 데이터 모델
- (2) 데이터베이스를 2차원 테이블 형태로 구성한 모델
- (3) 기본키와 이를 참조하는 외래키로 데이터 간의 관계를 표현한다.
- (4) 1:1, 1:N, M:N관계를 자유롭게 표현할 수 있다.

계층 데이터 모델



- (1) 데이터베이스의 논리적 구조가 트리(tree) 형태
- (2) 사이클이 허용되지 않음
- (3) 개체 간에 상하 관계가 성립되며 관계를 부모-자식 형태로 표현
 - 부모와 자식 개체는 일대다(1 : n) 관계

네트워크 데이터 모델



- (1) 데이터베이스의 논리적 구조 표현을 그래프 형태로 표현
- (2) 개체 간에는 일대다(1 : n) 관계만 허용됨
 - 오너(owner) / 멤버(member)
- (3) 상위 하나의 레코드에 대하여 하위의 레코드가 복수 대응,
하위 하나의 레코드에 대하여 상위 레코드가 복수 대응된다.
상위와 하위 레코드가 다대다의 대응관계 가능

<정리>

데이터 모델링 과정을 세단계로 나누어서 설명하시오.

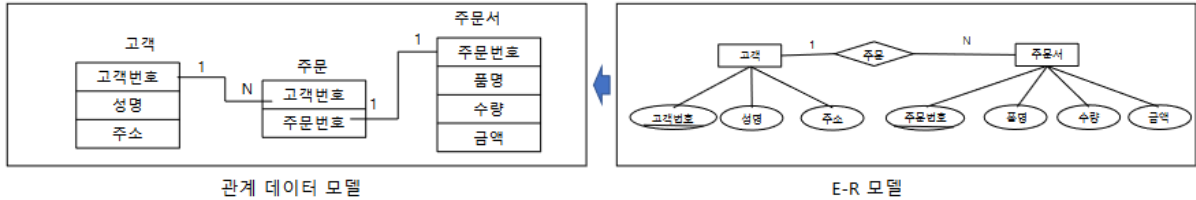
<평가>

1. 논리적 데이터 모델에 대한 설명으로 옳지 않은 것은? [1]
 - ① 개체관계 모델은 개체와 개체 사이의 관계성을 이용하여 데이터를 모델링한다.
 - ② 관계형 모델은 논리적 데이터 모델에 해당한다.
 - ③ SQL은 관계형 모델을 따르는 DBMS의 표준 데이터 언어이다.
 - ④ 네트워크 모델, 계층 모델은 레거시 데이터 모델로도 불린다.
2. E-R 다이어그램의 구성 요소에 대한 표현이 틀리게 짝지어진 것은? [4]
 - ① 개체 집합-직사각형
 - ② 관계 집합-마름모꼴
 - ③ 속성-원
 - ④ 링크-화살표

<5주차 : 데이터베이스 모델링>

1. 관계 데이터 모델의 개념

개념

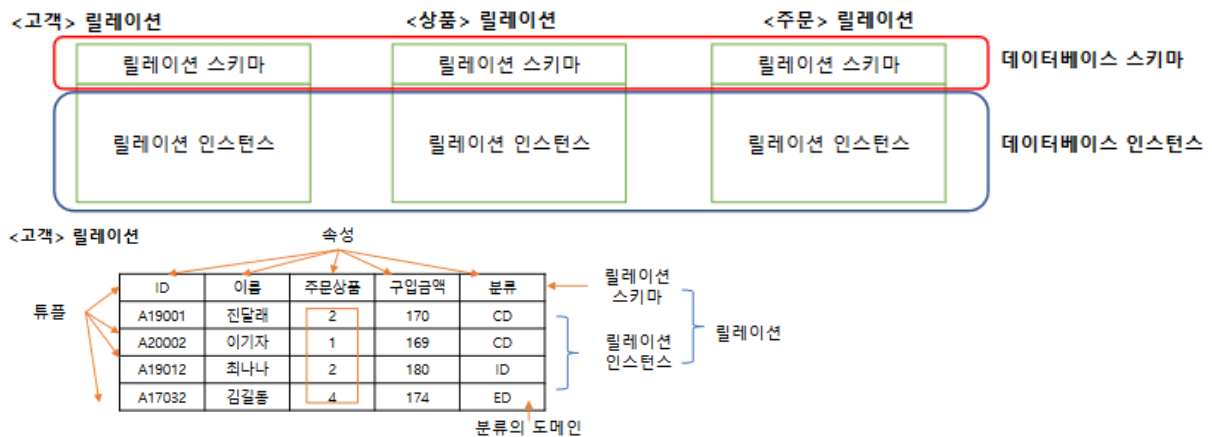


- (1) 개념적 구조를 논리적 구조로 표현하는 논리적 데이터 모델
- (2) 가장 널리 사용되는 데이터 모델로, 2차원적인 표(Table)를 이용해서 데이터 상호 관계를 정의.
- (3) 기본키(Primary Key)와 이를 참조하는 외래키(Foreign Key)로 데이터 간의 관계를 표현
- (4) 1:1, 1:N, N:M 관계를 자유롭게 표현할
- (5) 하나의 개체에 관한 데이터를 하나의 릴레이션에 저장

고객	주문	주문서
고객번호	고객번호	주문번호
성명	주문번호	품명
주소		수량
		금액
001	001	101
002	001	102
003	002	103
004	002	104
	003	105
	003	106
		107
		108

- (1) <고객> 테이블과 <주문서> 테이블은 <주문> 테이블을 관계로 하여 연결되어 있다.
- (2) <고객> 테이블의 '고객번호'와 <주문서> 테이블의 '주문번호'는 기본키
- (3) <주문> 테이블의 '고객번호'와 '주문번호'는 외래키
- (4) <고객>과 <주문>의 관계는 1:N, 즉 한 명의 고객은 여러 개의 주문 가능
- (5) <주문서>와 <주문>의 관계는 1:1, 즉 '주문서'는 '주문번호' 1개에 대한 정보만을 가지고 있다.

데이터베이스의 구성



- (1) 데이터베이스 스키마(database schema)
 - 데이터베이스의 전체 구조, 데이터베이스를 구성하는 릴레이션 스키마의 모음

(2) 데이터베이스 인스턴스(database instance)

- 데이터베이스를 구성하는 릴레이션 인스턴스의 모음

릴레이션의 구성

(1) 릴레이션(relation)

- 하나의 개체에 관한 데이터를 2차원 테이블의 구조로 저장한 것
- 파일 관리 시스템 관점에서 파일(file)에 대응

(2) 릴레이션 스키마(relation schema)

- 릴레이션의 논리적 구조
- 릴레이션의 이름과 릴레이션에 포함된 모든 속성의 이름으로 정의
- <예시> 고객(고객아이디, 고객이름, 나이, 등급, 직업, 적립금)
- 릴레이션 내포(relation intension)라고도 함, 자주 변하지 않는 정적인 특징

(3) 릴레이션 인스턴스(relation instance)

- 어느 한 시점에 릴레이션에 존재하는 튜플들의 집합
- 릴레이션 외연(relation extension)이라고도 함
- 튜플의 삽입, 삭제, 수정이 자주 발생하는 동적인 특징

기본용어

(1) 튜플(tuple)

- 릴레이션에서 하나의 행(Row)을 의미
- 파일 관리 시스템 관점에서 레코드(record)에 해당
- 기수(cardinality) : 하나의 릴레이션에서 튜플의 전체 개수(행의 수)

(2) 속성(attribute)

- 데이터베이스를 구성하는 가장 작은 논리적 단위
- 릴레이션에서 열(column)의 이름
- 파일 관리 시스템 관점에서 필드(field)에 해당
- 차수(degree) : 하나의 릴레이션에서 속성의 전체 개수(열의 수)

(3) 도메인(domain)

- 하나의 속성이 가질 수 있는 모든 값들의 집합
- 도메인은 실제 속성값이 나타날때 그 값의 합법 여부를 시스템이 검사하는데 이용
- <예시> 성별의 도메인 : 남, 여

릴레이션의 특성

(1) 튜플의 유일성

- 하나의 릴레이션에는 동일한 튜플이 존재할 수 없다.

(2) 튜플의 무순서

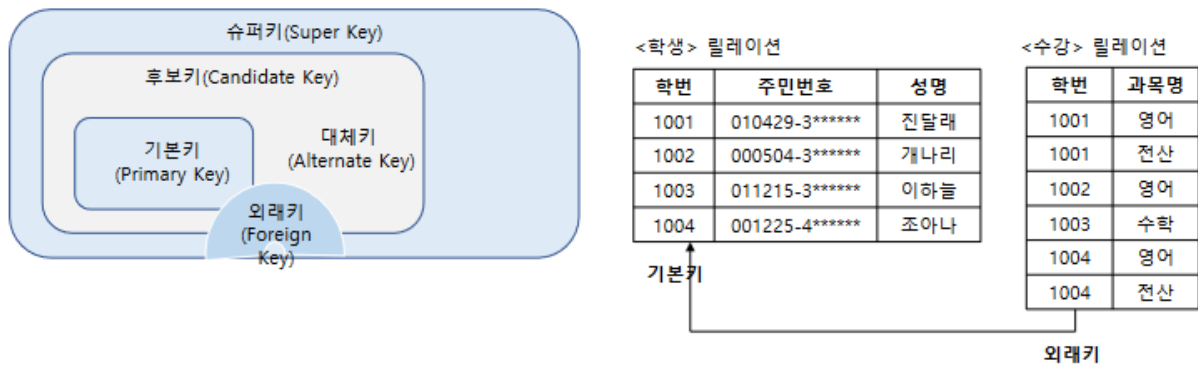
- 하나의 릴레이션에서 튜플 사이의 순서는 무의미하다.

(3) 튜플들의 삽입, 삭제 등의 작업으로 인해 릴레이션은 시간에 따라 변한다.

(4) 속성의 무순서

- 하나의 릴레이션에서 속성 사이의 순서는 무의미하다.
- (5) 속성의 원자성
- 속성 값은 논리적으로 더 이상 쪼갤 수 없는 원자값만을 저장한다.
- (6) 속성의 유일한 식별을 위해 속성의 명칭은 유일해야 하지만, 속성을 구성하는 값은 동일한 값이 있을 수 있다.
- (7) 릴레이션을 구성하는 튜플을 유일하게 식별하기 위해 속성들의 부분집합을 키(Key)로 설정한다.

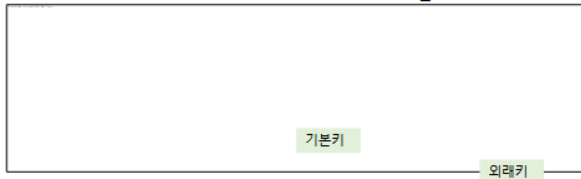
키(KEY)



- 데이터베이스에서 조건에 만족하는 튜플을 찾거나 순서대로 정렬할 때, 튜플을 서로 구분할 수 있는 기준이 되는 속성

키의 종류

- (1) 슈퍼키(super key)
 - ✓ 튜플을 유일하게 구분하는 키로서 1개 이상의 속성으로 구성
 - ✓ 유일성만 보장하는 속성으로 여러 개 존재 가능
- (2) 후보키(candidate key)
 - 유일성과 최소성을 만족하는 속성 또는 속성들의 집합
 - 모든 릴레이션에는 반드시 하나 이상의 후보키가 존재
- (3) 기본키(primary key)
 - 후보키 중에서 기본적으로 사용하기 위해 선택된 하나의 키
 - 한 릴레이션에서 특정 튜플을 유일하게 구별할 수 있는 속성
 - 중복된 값을 가질 수 없다.
 - 기본키는 NULL을 가질 수 없다.
- (4) 대체키(alternate key)
 - 후보키 중에서 기본키를 제외한 나머지 키 10
- (5) 외래키(Foreign Key)



- 다른 릴레이션의 기본키를 참조하는 속성.
- 외래키 값은 참조된 릴레이션의 속성에 존재하는 값이거나 NULL 값이어야 한다.
(외래키값은 참조된 릴레이션의 속성에 없는 값은 입력할 수 없다.)
- 외래키 속성과 그것이 참조하는 기본키 속성의 이름은 달라도 되지만 도메인은 같아야 한다.
- 하나의 릴레이션에는 외래키가 여러 개 존재할 수도 있고 외래키가 기본키로 될 수 있다.
- 릴레이션들 사이의 관계를 표현
 - 참조하는 릴레이션 : 외래키를 가진 릴레이션
 - 참조되는 릴레이션 : 외래키가 참조하는 기본키를 가진 릴레이션

키의 특성

- (1) 유일성(uniqueness)
 - 하나의 릴레이션에서 모든 튜플은 서로 다른 키 값을 가져야 함
- (2) 최소성(minimality)
 - 꼭 필요한 최소한의 속성들로만 키를 구성함

종류	<ul style="list-style-type: none"> • 수퍼키 : 유일성을 만족하는 속성 • 후보키 : 유일성과 최소성을 만족하는 속성 • 기본키 : 후보키 중에서 기본적으로 사용하기 위해 선택한 키 • 대체키 : 후보키 중에서 기본키로 선택되지 못한 후보키 • 외래키 : 다른 릴레이션의 기본키를 참조하는 속성
----	---

2. 관계 데이터 모델의 제약조건

제약조건

- (1) 도메인 제약 (Domain Constraints)
 - 속성(Attribute)에 대한 제약
- (2) 키 제약 (Key Constraints)
 - 릴레이션(Relation)에 대한 제약
- (3) 개체 무결성 제약 (Entity Integrity Constraints)
 - 기본키(Primary Key)에 대한 제약
- (4) 참조 무결성 제약 (Referential Integrity Constraints)
 - 외래키(Foreign Key)에 대한 제약

[1] 도메인 제약 (Domain Constraints)

학생

학번	이름	나이	차량번호	취미
91234	진달래	22	22로1234	필라테스
95678	최나나	대한		수영, 골프, 스쿼시

1

2

3

- (1) 속성 값은 원자성(atomicity)을 가지며, 도메인에서 정의된 값이어야 함
- (2) 복합속성(Composite Attribute)과 다중값속성(Multivalued Attribute)은 허용되지 않음
 - cf) 주소 = 시군구 + 상세주소
- (3) Null 값은 허용됨 (Not Null이 아닌 경우)

[2] 키 제약 (Key Constraints)

1

이름	나이	혈액형	전공
진달래	22	A	데이터사이언스
최나나	23	O	정보시스템

2

이름	나이	학번	주민번호
진달래	22	91234	010215-2363213
최나나	22	95678	010216-1363213

- (1) 릴레이션의 모든 튜플(Tuples)은 서로 식별 가능해야 함
- (2) (Q) 다음 중 키 제약을 위반하고 있는 릴레이션은?

[3] 개체 무결성 제약 (Entity Integrity Constraints)

학생

학번	이름	나이	차량번호
91234	진달래	22	01가1234
95678	최나나	23	
	김하나	23	
91234	유유이		02나3456

- 1
- 2
- 3
- 4

- (1) 기본키 (PK – Primary Key) 는 NOT NULL & UNIQUE 이어야 함
- (2) (Q) 다음 중 개체 무결성 제약에 위배되지 않는 튜플은?

[4] 참조 무결성 제약 (Referential Integrity Constraints)

학생(R1)

학번	이름	나이	소속	멘토
91234	진달래	22	DC	
92345	최나나	23	DC	1234
93456	김하나	23	경영	
94567	유유이	22	컴공	2345

학과(R2)

학과명	정원	위치
DC	50	백주년관
경영	60	경상관
컴공	50	공학관
수학	40	자연관

기본키

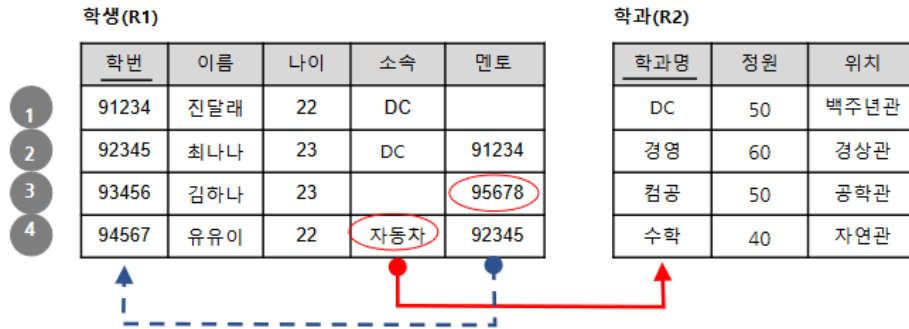
외래키

외래키

기본키

(1) 외래키 (FK – Foreign Key)

- 릴레이션 R1이 릴레이션 R2를 참조하는 경우, R2의 기본키는 R1에서 외래키로 사용됨
- FK는 자기 자신이 속한 릴레이션을 참조할 수도 있음



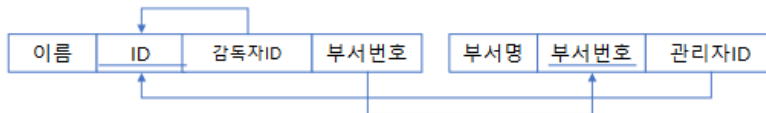
(2) 릴레이션 R1 이 릴레이션 R2를 참조하는 경우, R1 의 FK는...

- (1) Null 이거나
- (2) Null 이 아닌 경우 R2에 실제로 존재하는 값으로 구성되어야 함

(3) (Q) 다음 중 참조 무결성을 위배하는 튜플은?

제약조건

(1) 스키마



(Q) 다음 연산들은 어떤 문제를 야기하는가?

직원

이름	ID	감독자ID	부서번호
강감찬	A111	A222	5
김유신	A222	A888	5
이성계	A333	A444	4
이순신	A444	A888	4
정몽주	A555	A222	5
최무선	A666	A222	5
최지원	A777	A444	4
홍길동	A888	NULL	1

부서

부서명	부서번호	관리자ID
개발팀	5	A222
인사팀	4	A444
기획팀	1	A666

- - 새 직원 <'유관순', '555', NULL, 4> 삽입
- - 새 직원 <'이방원', '123', '456', 5> 삽입
- - ID='A666' 인 직원 삭제

무결성(Integrity)

- 데이터베이스에 저장된 데이터 값과 그것이 표현하는 현실 세계의 실제값이 일치하는 정확성을 의미
- 무결성 제약 조건 : 데이터베이스에 들어 있는 데이터의 정확성을 보장하기 위해, 부정확한 자료가 데이터베이스 내에 저장된 것을 방지하기 위한 제약 조건

데이터 무결성 강화

- (1) 데이터 무결성은 데이터 품질에 직접적인 영향을 미치므로, 데이터 특성에 맞는 적절한 무결성을 정의하고 강화해야 한다.
- (2) 프로그램이 완성되고 데이터가 저장된 상태에서 무결성을 정의할 경우 많은 비용이 발생하므로, 데이터베이스 구축 과정에서 정의한다.
- (3) 데이터 무결성은 애플리케이션, 데이터베이스 트리거, 제약 조건을 이용하여 강화할 수 있다.

(1) 애플리케이션

- 데이터를 생성, 수정, 삭제할 때 무결성 조건을 확인하는 코드를 데이터 조작 프로그램에 추가
- <예시> 데이터 유효성검사, 특정규칙 적용
- <장점> 데이터 유효성 검사를 맞춤형으로 설계 가능, 사용자 정의와 같은 복잡한 무결성 조건의 구현이 가능
- <단점> 데이터 유효성 검사를 여러 곳에서 구현하므로 중복성이 발생할 수 있고 유지보수의 어려움

(2) 데이터베이스 트리거

: 데이터 조작 이벤트(예: INSERT, UPDATE, DELETE)가 발생할 때 자동으로 실행

- 특정 조건을 충족하지 않는 데이터 조작을 거부하거나, 데이터 조작 시 일관성을 검사
- <장점> 데이터 조작 이벤트 발생 시 트리거가 자동으로 실행되므로 개발자가 별도의 코드를 작성할 필요가 없음
- <단점> 운영 중 변경이 어렵고, 사용상 주의가 필요하다.

(3) 제약 조건

- 데이터베이스에 제약 조건을 설정하여 무결성을 유지
- <장점> 통합관리 가능, 간단한 선언으로 구현가능, 변경 용이, 오류 데이터 발생방지 등이 있다.
- <단점> 복잡한 제약 조건의 구현과 예외적인 처리가 불가능하다.

<정리>

(1) 관계 데이터베이스

이름	ID	감독자ID	부서번호
강감찬	A111	A222	5
김유신	A222	A888	5
이성계	A333	A444	4
이순신	A444	A888	4
정몽주	A555	A222	5
최무선	A666	A222	5
최치원	A777	A444	4
홍길동	A888	NULL	1

- Relation ()
- Degree ()
- Cardinality ()

(2) <직원> 릴레이션의 기본키는 '직원번호', <지점> 릴레이션은 기본키가 미설정된 상태이다. <지

점>릴레이션의 후보키인 지점번호를 기본키로 설정하면 참조 무결성 제약조건과 개체 무결성 제약조건에 위배된다. 두가지 무결성 제약조건의 위배사유를 설명하시오.

<직원>

직원번호	주민번호	이름	지점번호
A1	A111	A222	101
A2	A222	A888	103
A3	A333	A444	102
A4	A444	A888	106
A5	A555	A222	101

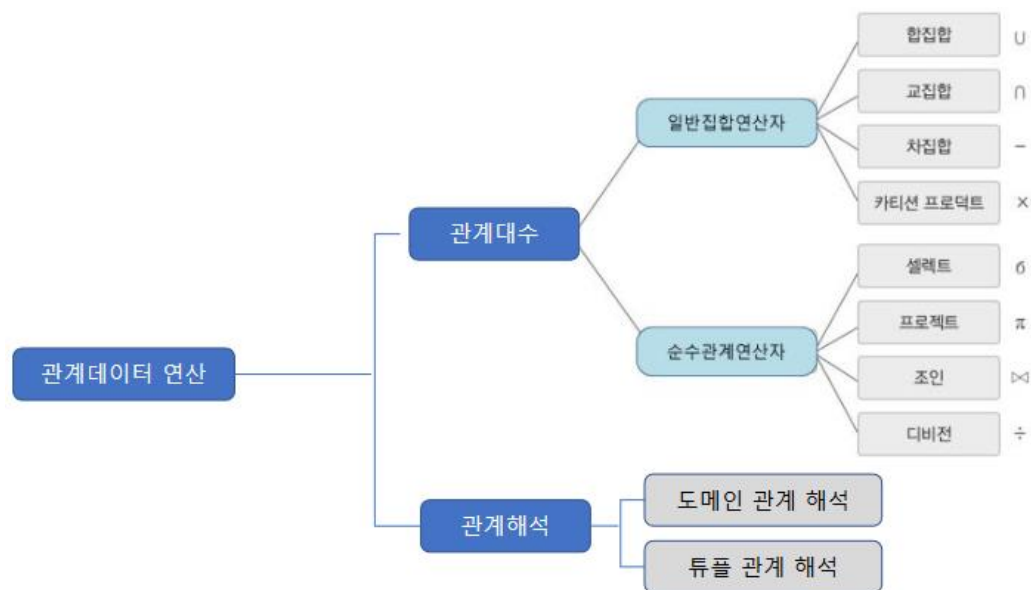
<지점>

지점번호	주소	직원수
101	서울	22
103	서울	88
NULL	부산	44
106	광주	88
104	광주	22

- 참조무결성 위배사유
- 개체무결성 위배사유

<6주차 : 관계데이터 연산>

1. 관계데이터 연산



개념

(1) 개념적 구조를 논리적 구조로 표현하는 논리적 데이터 모델

2. 관계 대수

개념

(1) 개념적 구조를 논리적 구조로 표현하는 논리적 데이터 모델

3. 관계 해석

개념

(1) 개념적 구조를 논리적 구조로 표현하는 논리적 데이터 모델

