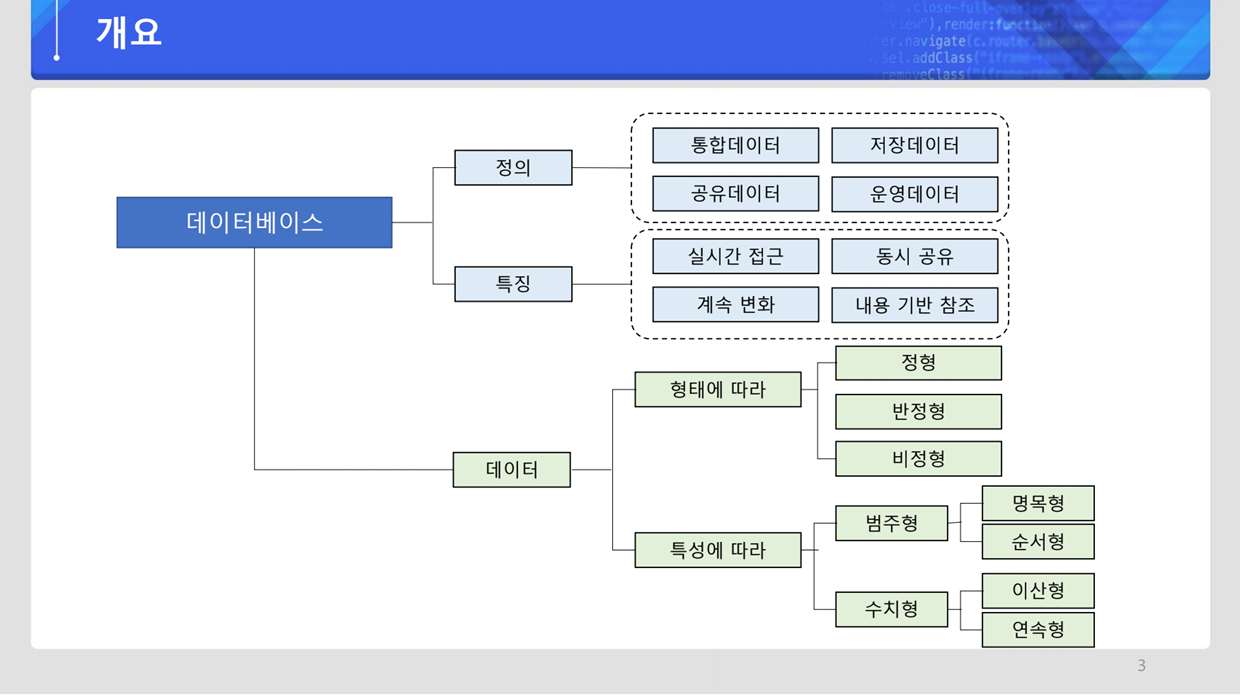
**<1주차 : 데이터베이스 기본 개념>**

**0. 개요**



**1. 데이터베이스 필요성**

\* 홈페이지마다 비밀번호를 전부 다르게 설정(or 몇 개월마다 교체)하는 것이 원칙. But,기억을 못해서, 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.

**# 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)

(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.

[ 데이터는 원시적이고 단순한 사실이나 값들의 집합이며, 정보는 이러한 데이터를 의미 있게 변환한 것. 정보는 데이터보다 높은 수준의 조작, 해석 또는 의미를 갖음. ]

(1) **기상 관측**

• **데이터** : 8월 18일, 기온: 30℃, 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℃, 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**

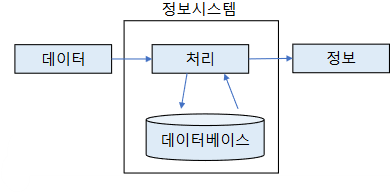
• 조직의 정보 처리 및 관리와 관련된 전반적인 시스템을 의미

• 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]

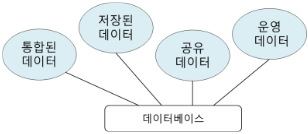
- 고객 관리 데이터베이스: 회사의 고객 정보, 주문 내역, 연락처 등을 저장하는 데이터베이스

- 도서관 데이터베이스: 도서의 정보, 대출 내역, 회원 정보 등을 저장하는 데이터베이스

**2. 데이터베이스 정의와 특징**

**# 데이터베이스 정의**

: 특정조직의 여러 사용자가 공유하여 사용할 수 있도록, 통합해서 저장한 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) **공용 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, 항고예약시스템, 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**

(1) **정형 데이터 (Structured Data)**

• 고정된 구조나 형식을 가진 데이터, 관계형 데이터베이스의 테이블과 같이 열과 행으로 이루어진 데이터

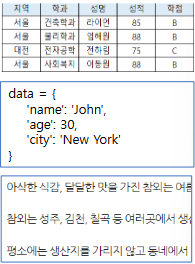
• [예시]

- 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**

(1) **범주형**

• 종류를 나타내는 값을 가짐. 연산 의미 없음

• 1) **명목형** : 서열이 없음(거주지역, 음식, 성별)

• 2) **순서형** : 서열이 있음(학년)

(2) **수치형**

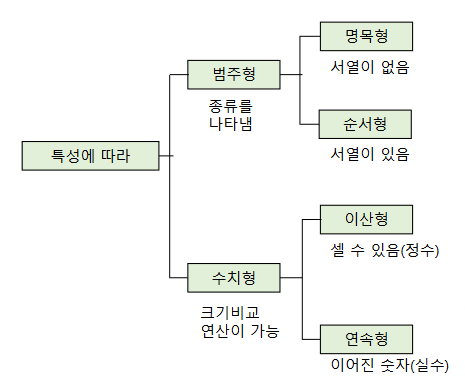
• 숫자로 표현되는 데이터, 크기비교와 산술연산이 가능

• [예시]

- 키, 몸무게

• 1) 이산형 : 셀 수 있음 (고객수, 판매량)

• 2) 연속형 : 연속적으로 이어진 숫자값 (키, 몸무게, 온도, 점수)



<정리>

• 데이터베이스의 정의에 대해 설명하시오.

• 데이터베이스의 네가지 특징에 대해 설명하시오.

• 데이터를 유형별로 분류하여 설명하시오.

<평가>

1. 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.

2. 다음 중 이산형 데이터에 속하는 것은? [3]

① 고객 성별

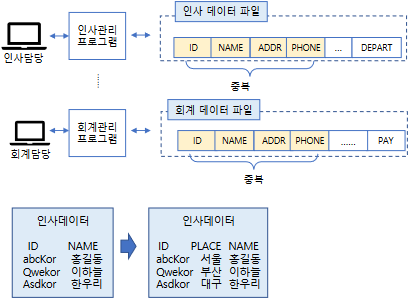
② 가입 기간

③ 가입 고객 수

④ 고객 거주 지역

**<2주차 : 데이터베이스 관리 시스템>**

**1. DBMS 개념**

**# 파일 시스템 (File System)**

(1) 중복 저장

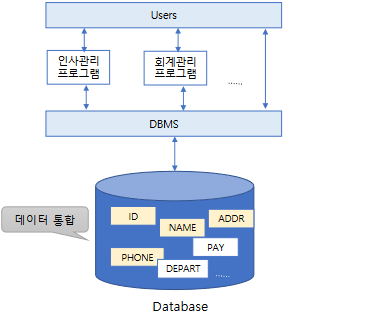
-> 저장 공간 낭비, 데이터의 일관성, 무결성 유지 어려움

(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.

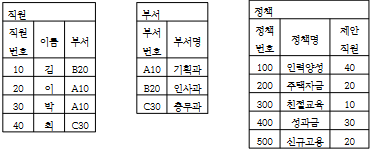
(2) 데이터를 데이터 베이스에 통합하여 저장하고 관리. (중복의 최소화)

✓ 중복성으로 인한 기존의 문제점

✓ 일관성, 보안성, 경제성, 무결성

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

**#### 질의(Query) 예시**



(1) “신규고용” 정책을 제안한 부서명? (검색)

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

(3) “인사과”에서 제안한 정책명을 삭제하세요. (삭제)



(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)

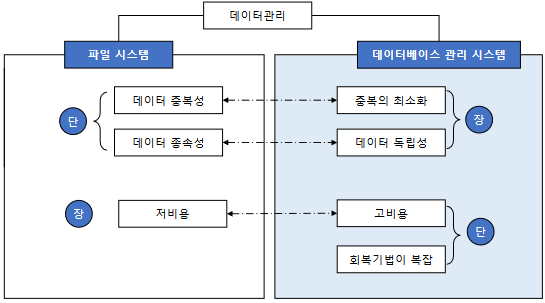
• 병행제어(여러 사용자 동시 접근 → 처리결과의 정확성유지)

• 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

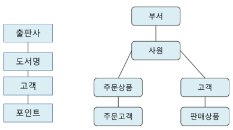
• 데이터 중복성, 데이터 무결성 문제, 데이터의 물리적 및 논리적 종속성 등의 문제

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

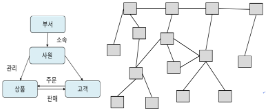


3. **네트워크형 DBMS(Network)**

• 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)

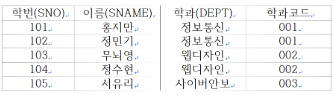
• 1970년, E.F. Codd에 의해 소개

• 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.



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

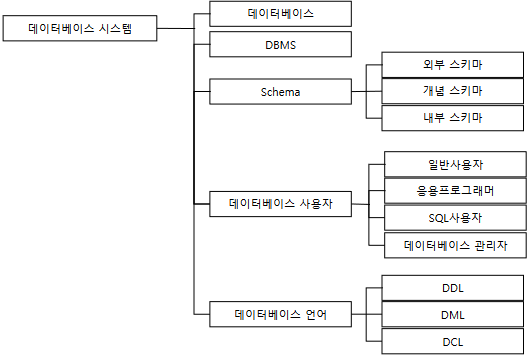
② 데이터 중복 최소화

③ 적은 비용

④ 데이터 독립성

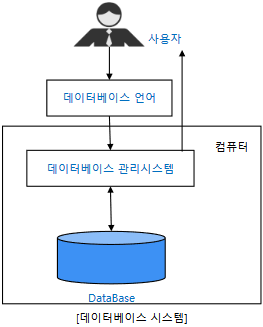
**<3주차 : 데이터베이스 시스템>**

**0. 정리**



**1. 데이터베이스 시스템 구성**

**# 데이터베이스 시스템(DBS, DataBase System)**

****: 데이터베이스에 데이터를 저장, 관리, 조직에 필요한 정보를 생성하는데 필요한 컴퓨터 중심 의 시스템

**## 데이터베이스 시스템 구성요소**

(1) 데이터베이스

(2) 데이터베이스 관리 시스템

(3) **스키마**

(4) **데이터베이스 사용자**

(5) **데이터베이스 언어**

**2. 스키마**

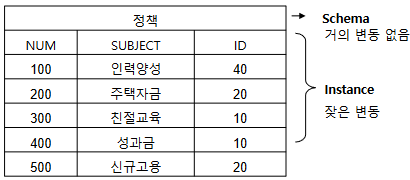
**# 스키마(schema)**

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

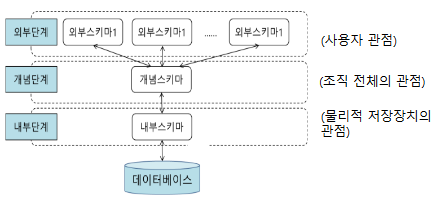


**## 인스턴스 (Instance)**

: 특정 시점에 데이터베이스에 실제로 저장되어 있는 데이터



**### 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

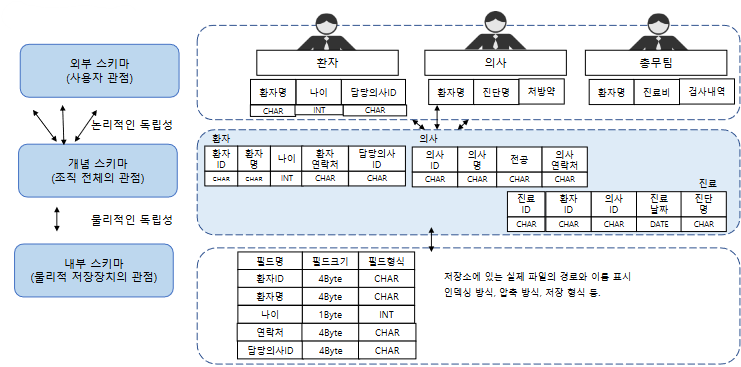
• 데이터베이스의 접근권한, 보안정책, 무결성 규칙에 관한 명세를 정의

(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.

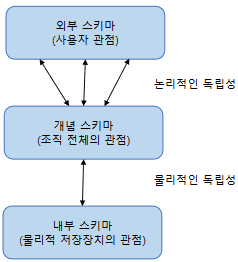
▪ [예시] 파일 시스템

- 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 하나만 존재한다.

④ 사용자나 응용 프로그래머가 접근하는 데이터베이스를 정의한 것이다.

**<4주차 : 데이터베이스 모델링>**

**1. 데이터 모델링**

**# 데이터 모델링 (data modeling)**

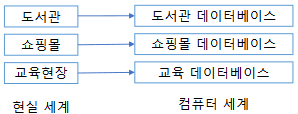
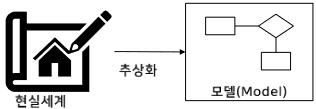
(1) 현실세계의 복잡한 정보와 관계를 컴퓨터 세계에서 처리할 수 있도록 변환하는 과정

(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

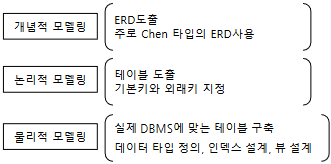
(2) 프로세스 관점(How)

• 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.

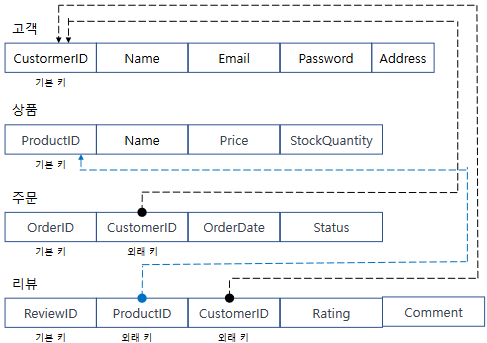
- 리뷰는 한 명의 고객에 의해 작성될 수 있다.

• 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.

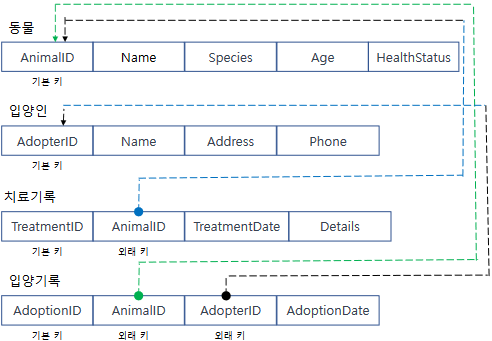
- 입양인은 여러 동물을 입양할 수 있다.

- 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)

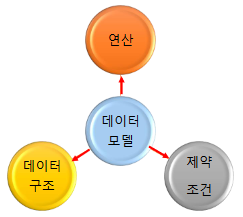
- 한 마리의 동물은 여러 치료 기록을 가질 수 있다. (동물 1:N 치료 기록)

- 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

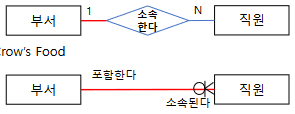
- 동적 특징

• 제약조건(constraint)

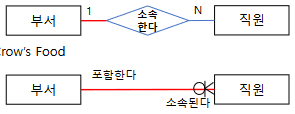
- 데이터 무결성 유지 목적

**##### 데이터 모델 표시법**

(1) Chen



(2) IE / Crow’s Food



(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

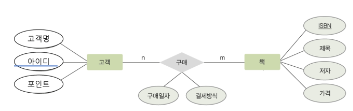
– Entity(개체): 데이터 모델에서 대상이나 개념

– 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) 엔터티를 그린 후 적절하게 배치

▪ 가급적 선이 꼬이지 않게 배치

▪ 왼쪽 → 오른쪽, 위 → 아래 순으로 읽어 나가기 편하도록 배치

(2) 엔터티간 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)**

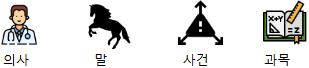
**### 엔터티(Entity)의 정의**

(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)



- 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

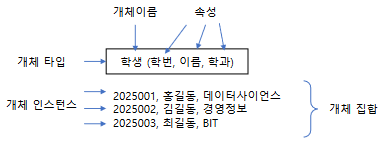
- 물리적인 형태는 존재하지 않으나 관리해야 할 개념적 정보

- 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 갖음

• 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

**3. 논리적 데이터 모델**

**# 데이터베이스 시스템(DBS, DataBase System)**

**<3주차 : 데이터베이스 시스템>**

**1. 데이터베이스 시스템 구성**

**# 데이터베이스 시스템(DBS, DataBase System)**