

소프트웨어공학



강의노트

09주차 02차시
아키텍처 참고모델 및 아키텍처 설계

❖ 학습안내

이번 시간의 학습내용과 학습목표를 확인해보세요.

■ 학습내용

- 응용시스템 아키텍처 참고모델
- 데이터베이스 설계
- 인터페이스 설계

■ 학습목표

- 응용시스템 참고모델을 활용하여 적절한 아키텍처를 설계할 수 있다.
- 데이터 베이스 구성을 설계할 수 있다.
- 인터페이스 구성을 설계할 수 있다.



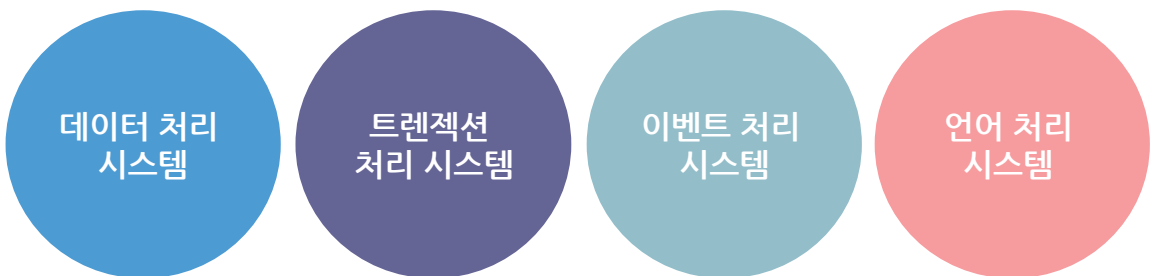
❖ 학습내용

[1] 응용시스템 아키텍처 참고모델

1. 데이터 처리 시스템

◆ 응용시스템 아키텍처 참고모델

- 시스템 하드웨어 구성 등에 따라 아키텍처를 분류하는 사례를 살펴보았음
- 어플리케이션(응용, 소프트웨어) 관점에서 **아키텍처를 분류**하여 몇몇 응용시스템 아키텍처에 대하여 살펴봄
- 응용시스템 아키텍처로는 다음과 같은 모델이 많이 사용됨



◆ 데이터 처리 시스템 개요

- 데이터 처리 응용 시스템은 **데이터의 처리 관리**를 중점으로 다루는 시스템
- 일반적으로 중앙에 데이터 베이스 등 **데이터 처리 시스템**을 두고 처리

예

급여지불시스템, 은행 계좌 시스템, 과금 시스템, 회계시스템 등 대부분의 **업무 처리 시스템**에서 광범위하게 사용됨

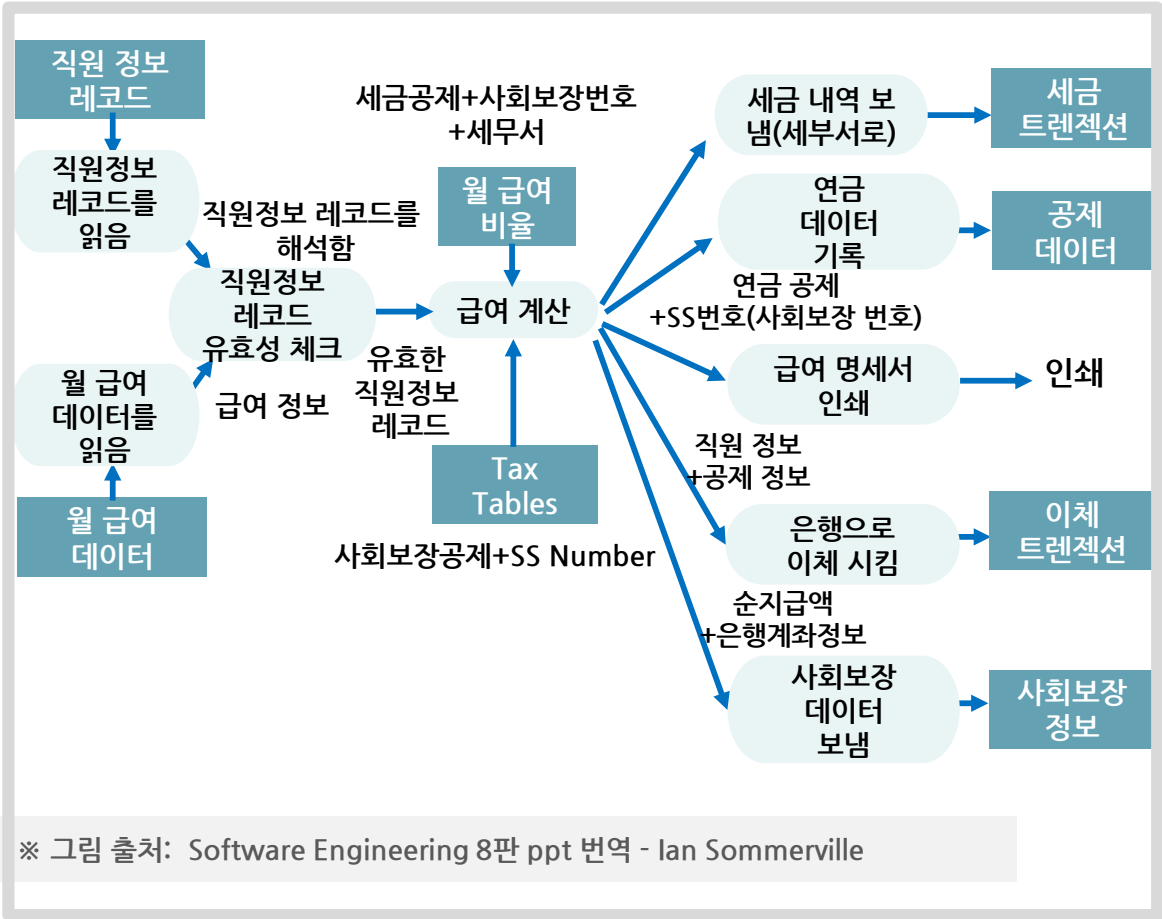
- 전통적인 업무처리 정보시스템 개념으로 지금까지도 대부분의 **기업의 정보시스템**이 이 부분에 해당됨

❖ 학습내용

[1] 응용시스템 아키텍처 참고모델

1. 데이터 처리 시스템 사례 - 미국의 일반적인 급여처리

◆ 다음은 데이터 처리 시스템의 하나인 일반적인 급여처리(미국의 사례)의 예제

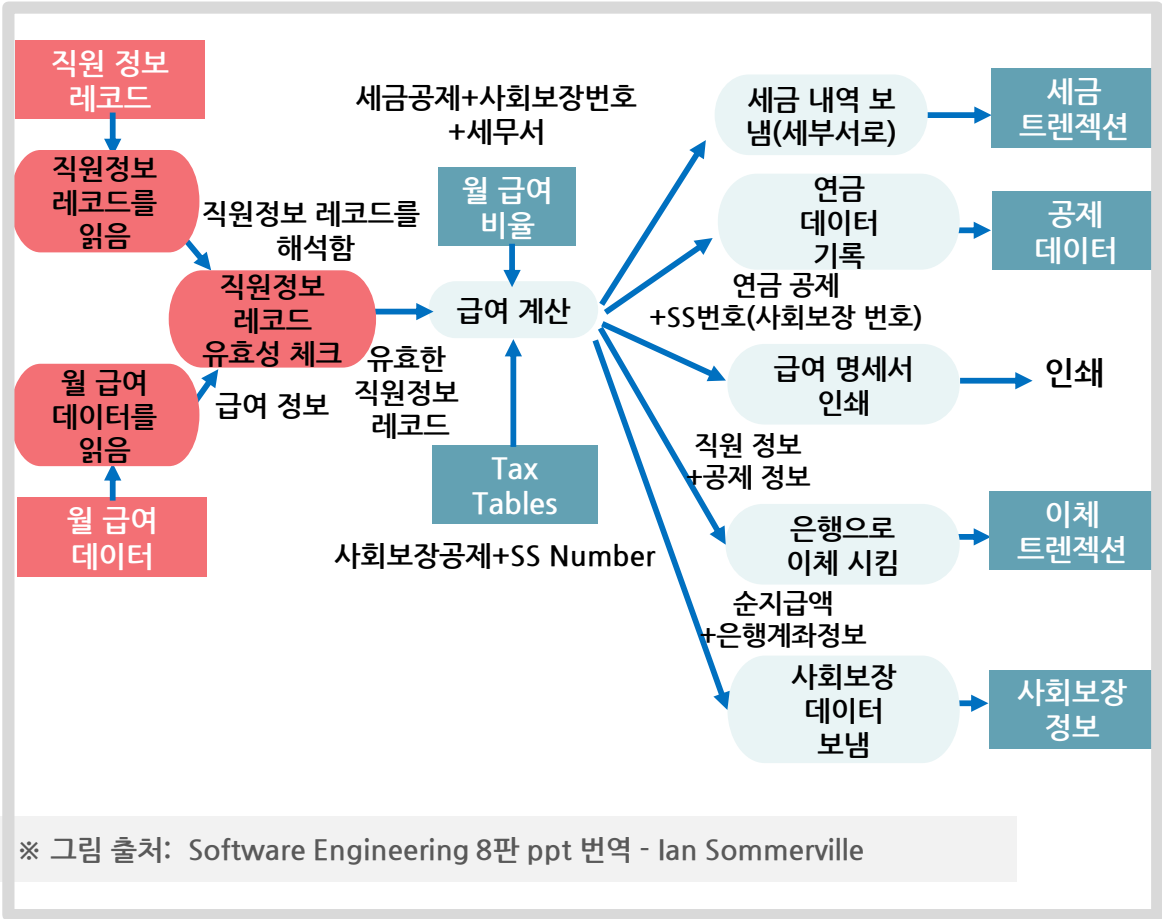


❖ 학습내용

[1] 응용시스템 아키텍처 참고모델

1. 데이터 처리 시스템 사례 - 미국의 일반적인 급여처리(계속)

◆ 직원 정보와 월 급여정보를 가지고 둘을 비교 체크함

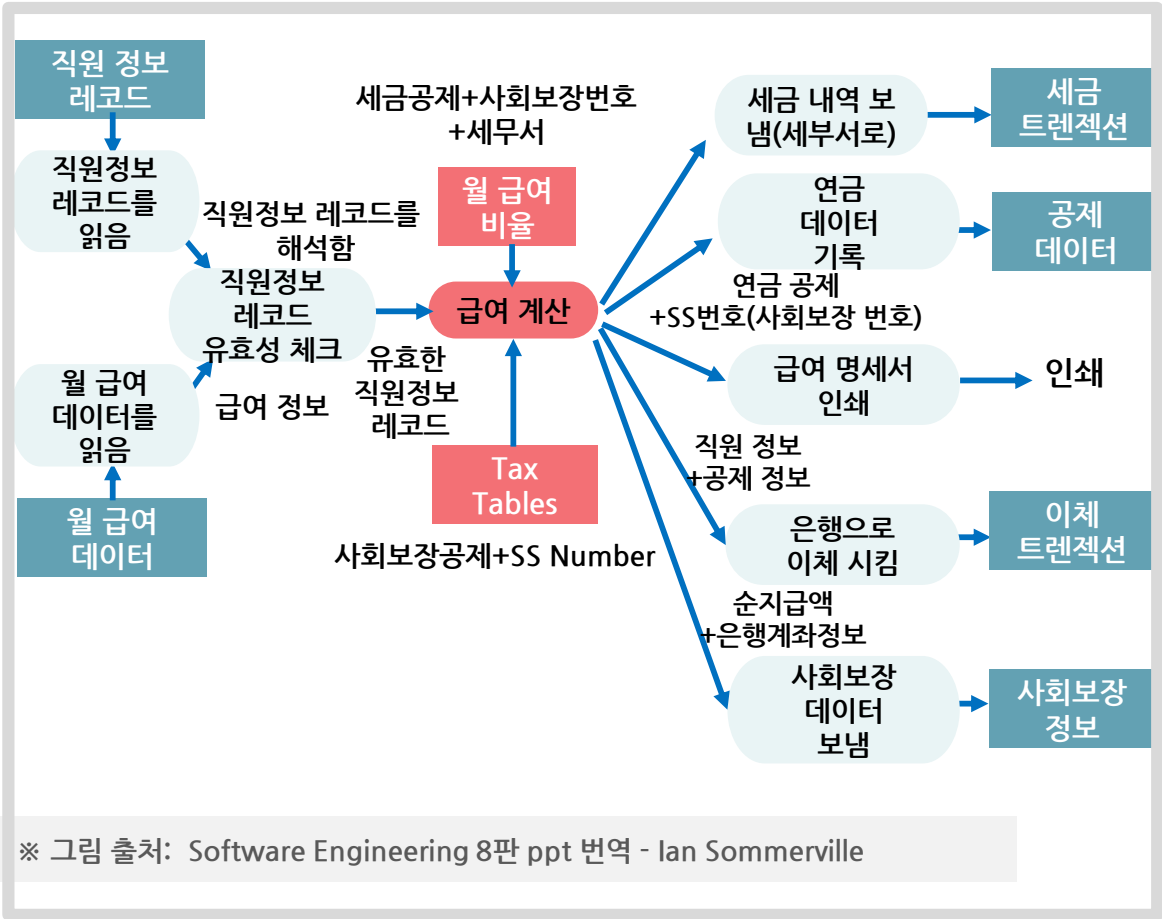


❖ 학습내용

[1] 응용시스템 아키텍처 참고모델

1. 데이터 처리 시스템 사례 - 미국의 일반적인 급여처리(계속)

- ◆ 월 급여율 정보와 세금테이블을 가지고 급여를 계산함
- ◆ 세금테이블에는 사회보장 보험 번호 등이 있어 연금 사회보장 보험금 등이 공제

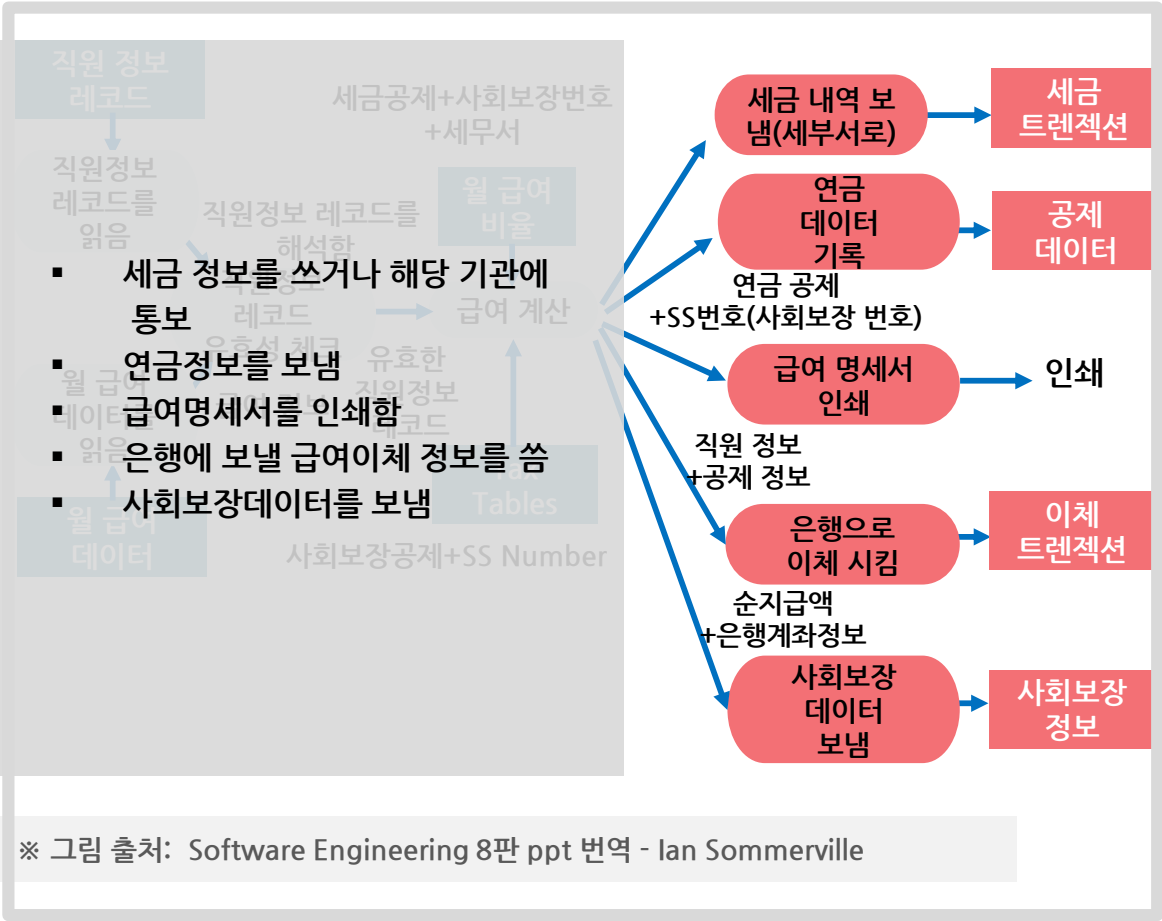


❖ 학습내용

[1] 응용시스템 아키텍처 참고모델

1. 데이터 처리 시스템 사례 - 미국의 일반적인 급여처리(계속)

◆ 다음은 데이터 처리 시스템의 하나인 일반적인 급여처리(미국의 사례)의 예제



❖ 학습내용

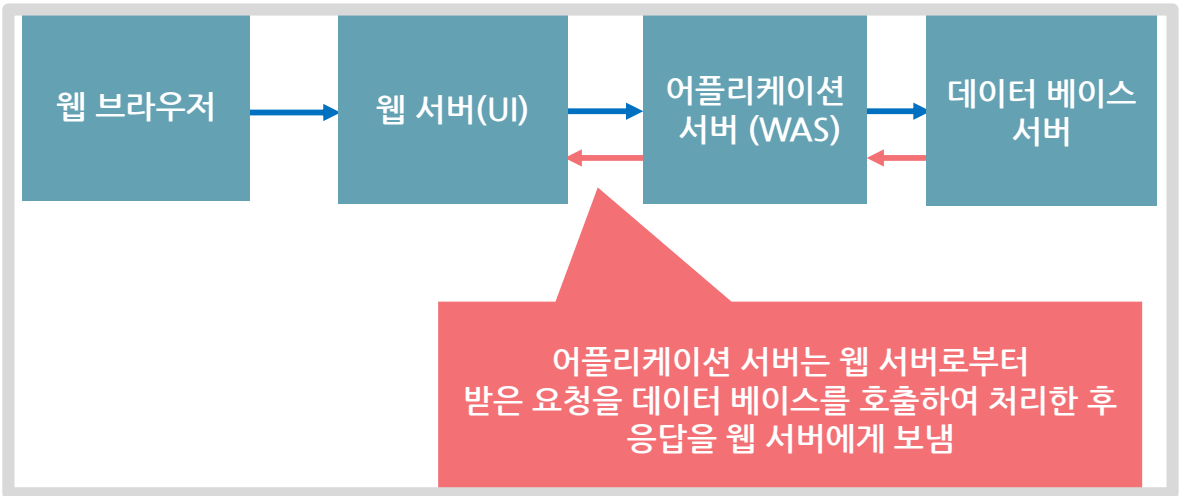
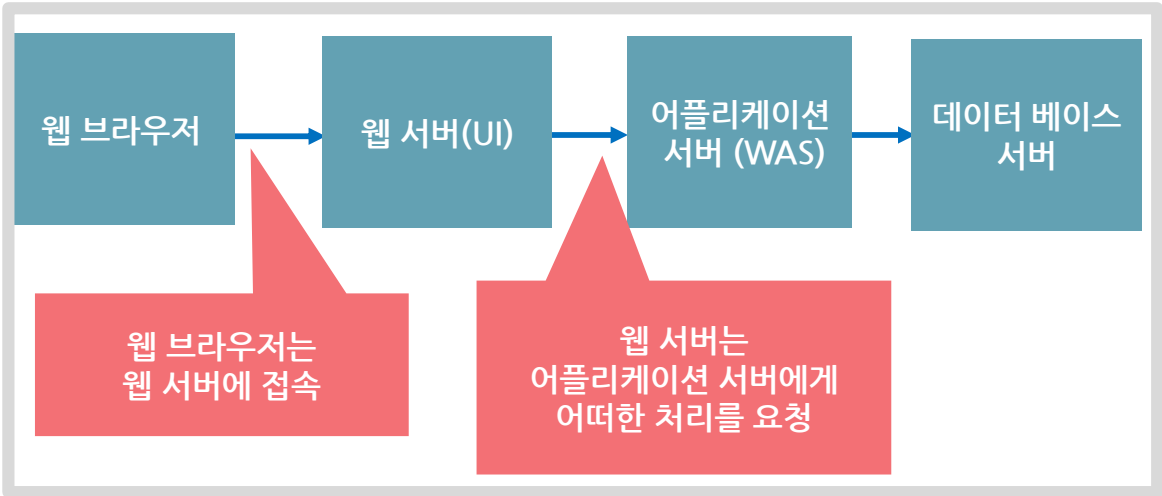
[1] 응용시스템 아키텍처 참고모델

2. 트랜잭션 처리 시스템

- ◆ 트랜잭션 처리 시스템 개요
 - 정보에 대한 요청에 대하여 처리하고, 정보를 갱신하는 등 **실시간 트랜잭션 처리 중심** 응용시스템
 - 일반적으로 **대화식 은행 시스템, 전자상거래 시스템, 정보시스템, 예약시스템**이 이에 해당
 - 웹 조회 방식처럼 어떠한 **정보의 요청**이 오면, 이에 대한 **결과의 응답**으로 이루어지는 시스템이 여기에 해당됨

2. 데이터 처리 시스템 사례 - 트랜잭션 처리 시스템

- ◆ 다음은 트랜잭션 처리 시스템의 하나인 일반적인 인터넷 쇼핑 시스템 구성도의 사례임

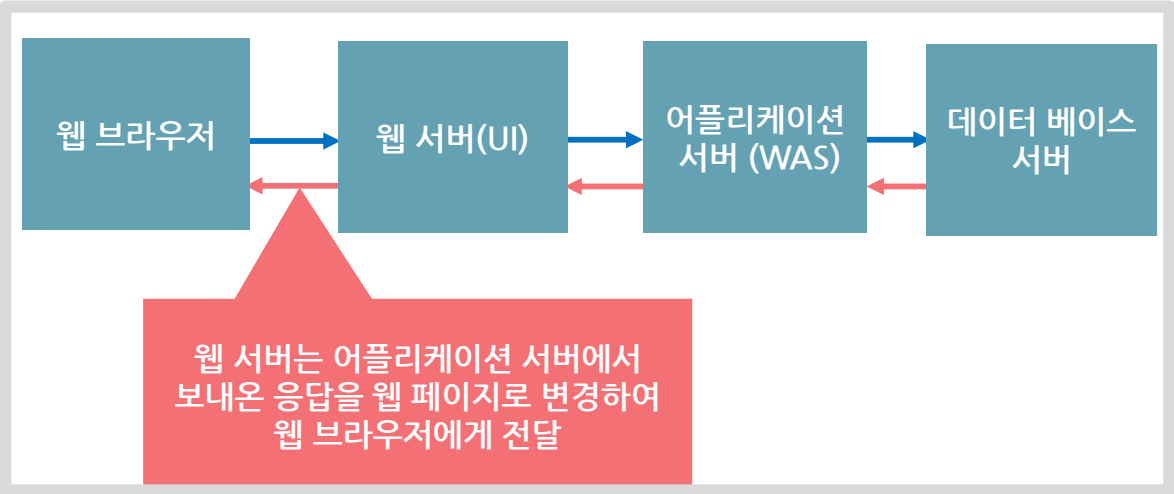


❖ 학습내용

[1] 응용시스템 아키텍처 참고모델

2. 데이터 처리 시스템 사례 - 트랜잭션 처리 시스템(계속)

◆ 다음은 트랜잭션 처리 시스템의 하나인 일반적인 인터넷 쇼핑 시스템 구성도의 사례임



3. 기타 시스템

◆ 이벤트 처리 시스템

- 이벤트를 해석하고 처리하는 방식의 시스템

예

- PC기반의 응용 시스템(마우스 감지, 키보드 감지)
- 실시간 시스템(주식매매, 레이더 탐지)

- 즉, 한쪽의 프로세스는 이벤트를 대기하고 있다가 이벤트가 감지되는 경우 어떤 처리를 하는 시스템
- 시스템 내부적으로 많은 이벤트가 처리되고 있음

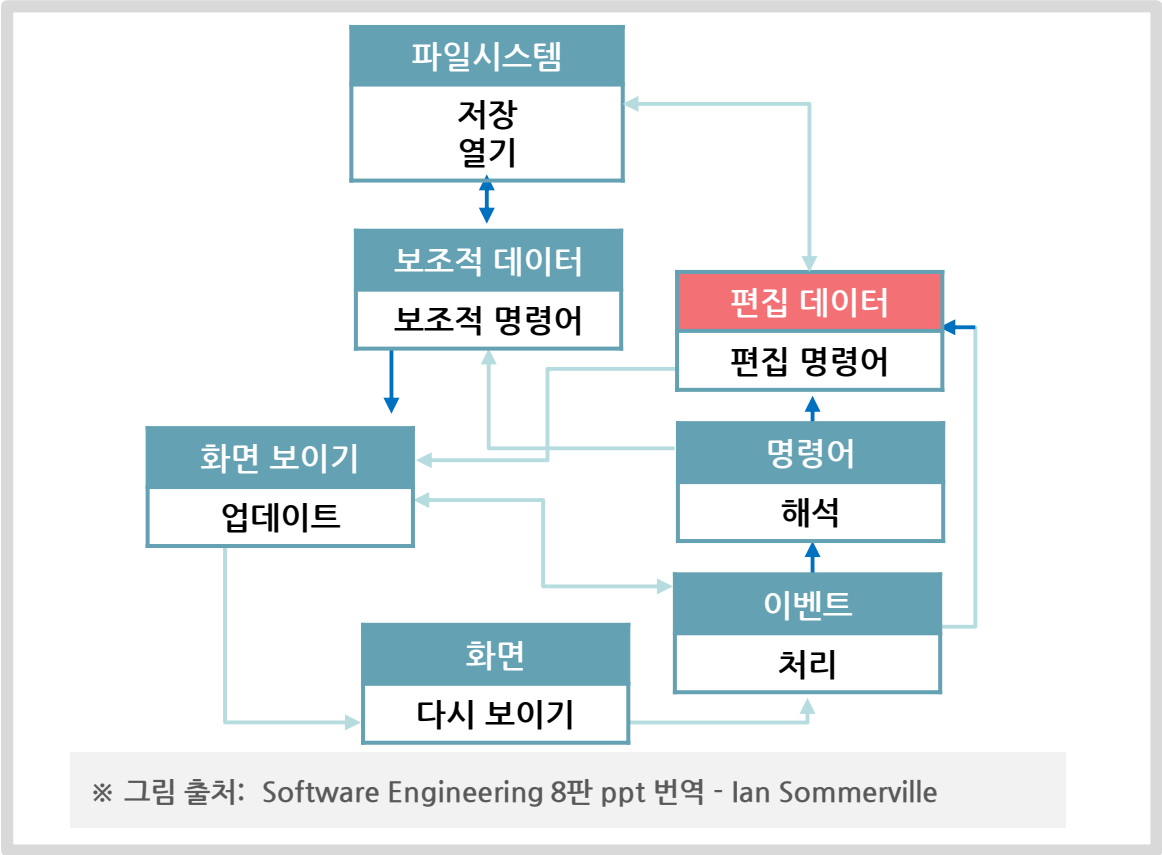
예

- 마우스가 움직였다. → 반응
- 손가락으로 화면을 터치했다. → 반응
- 온도센서의 온도가 100도에 도달하였다. → 물이 끓고 있다고 반응

❖ 학습내용

[1] 응용시스템 아키텍처 참고모델

3. 기타 처리 시스템 사례 - 이벤트 처리 시스템(계속)

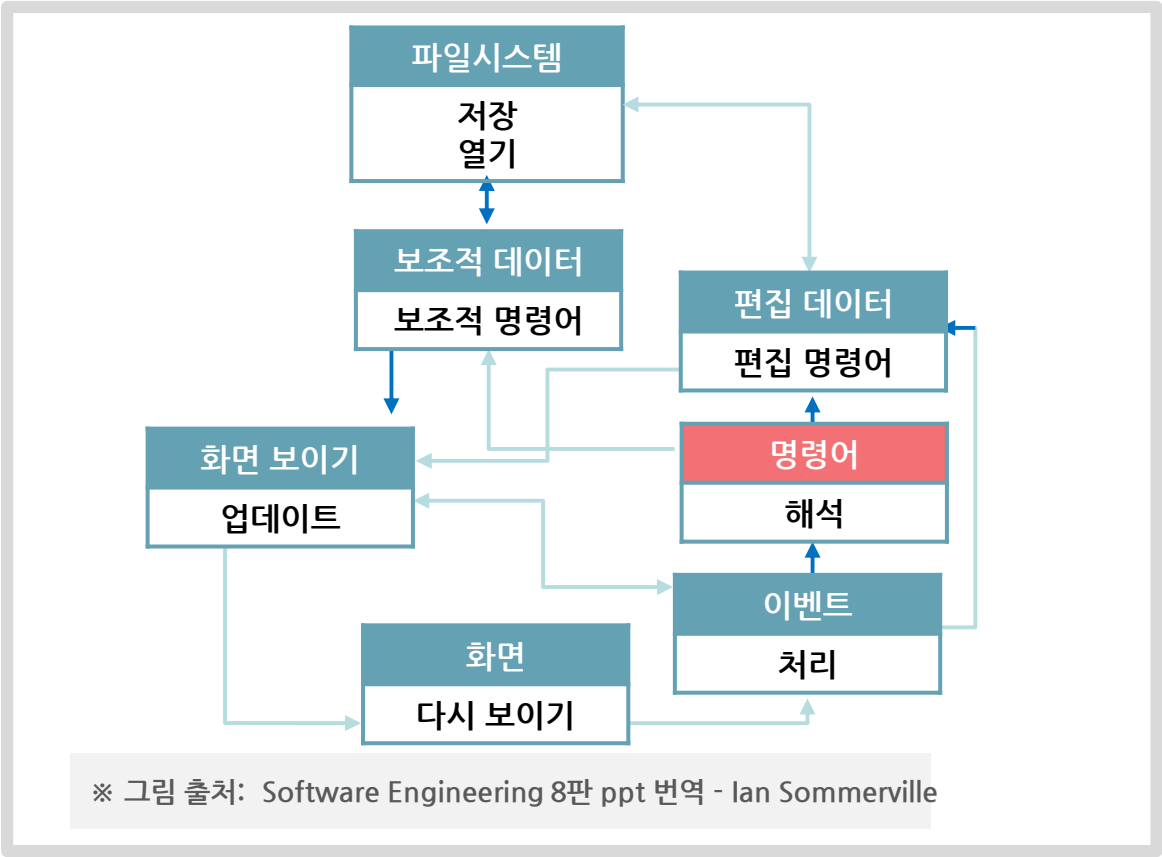


- ◆ 이벤트 처리 시스템의 하나인 **편집기(Editor, Word-processor)** 시스템의 **처리프로세스** 사례
- ◆ 항상 키보드의 **입력이벤트**를 받고 있음

❖ 학습내용

[1] 응용시스템 아키텍처 참고모델

3. 기타 처리 시스템 사례 - 이벤트 처리 시스템(계속)

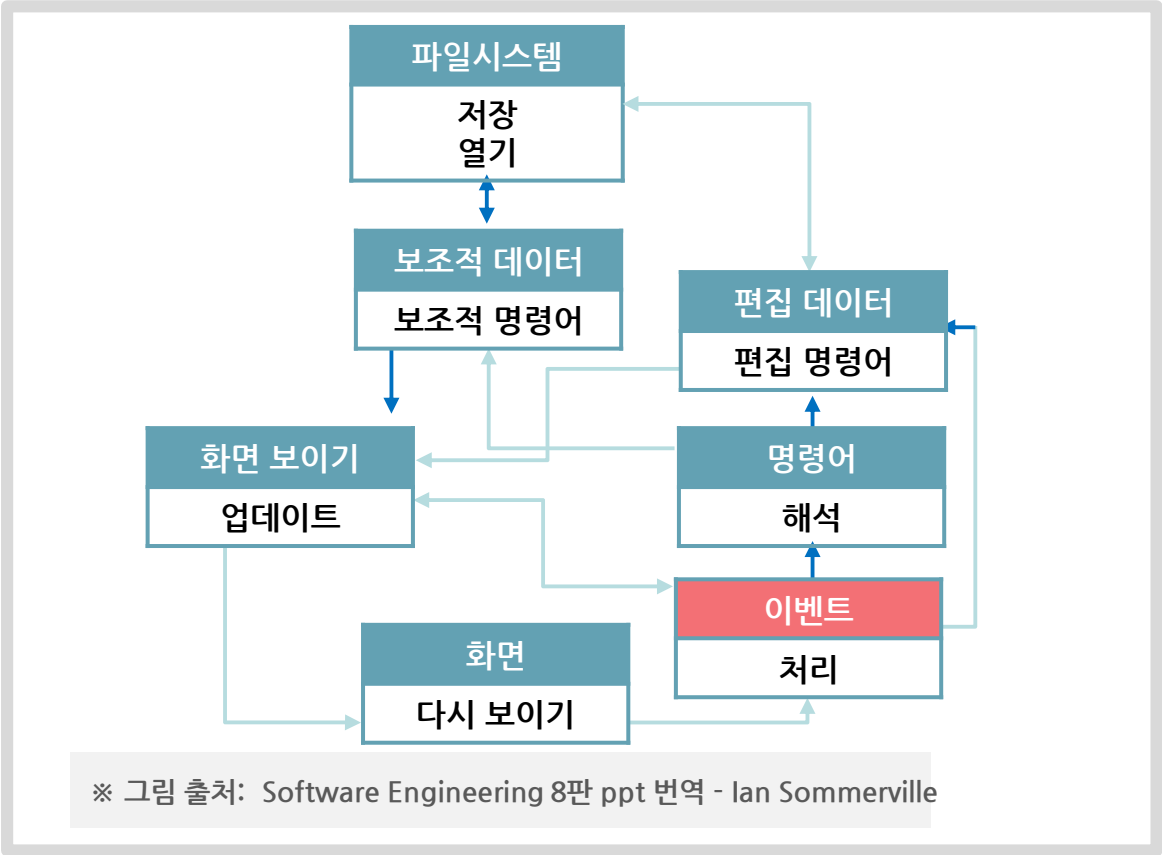


- ◆ 키보드나 마우스의 입력에 따라 이 입력이 글자를 쓰는 데이터(Editor Data)인지 명령인지, 부수적인(Ancillary) 데이터인지 부수적인 명령인지를 **판단하고 번역 (Interpret)**

❖ 학습내용

[1] 응용시스템 아키텍처 참고모델

3. 기타 처리 시스템 사례 - 이벤트 처리 시스템(계속)



- ◆ 해당 이벤트가 **명령이면 명령에 맞는 작업들을 수행**
- ◆ 해당 이벤트(키 입력)가 글자를 쓰는 작업이면 **끊임없이 화면을 다시 보여주는 작업을 계속 수행**

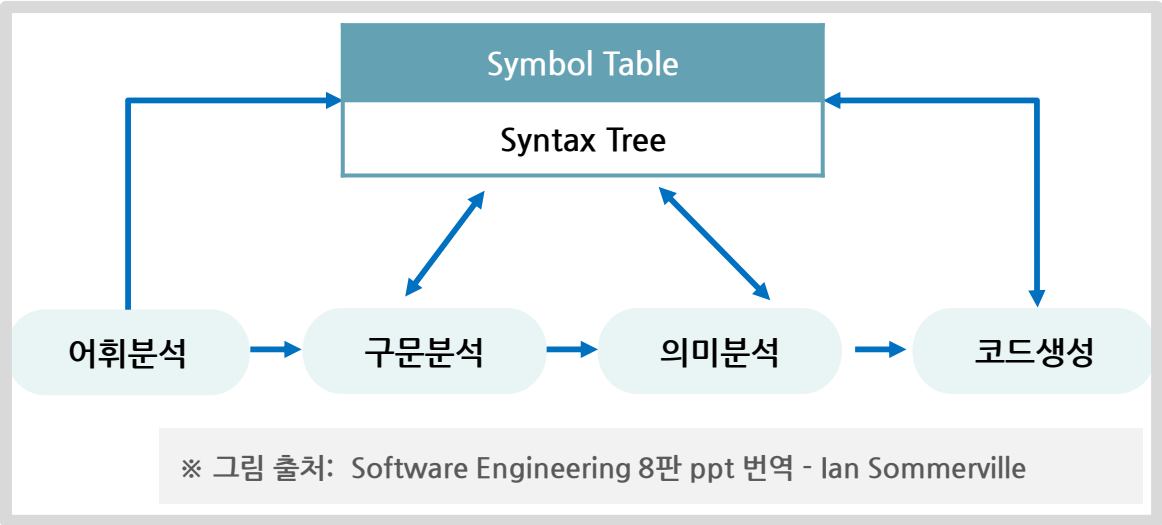
❖ 학습내용

[1] 응용시스템 아키텍처 참고모델

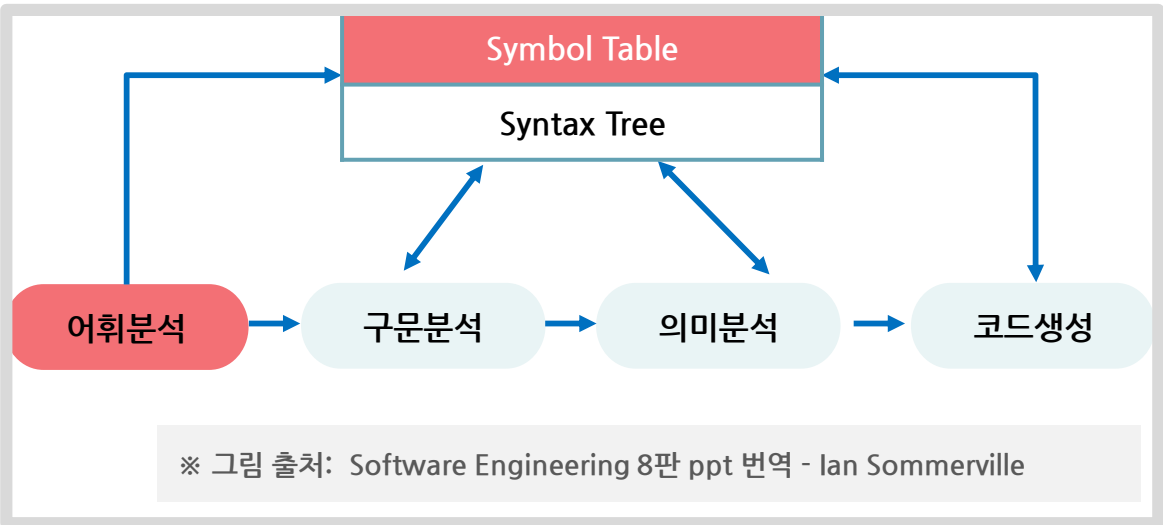
3. 기타 시스템

- ◆ 언어처리 시스템
 - 프로그램 언어의 처리 등을 하는 시스템으로 단순히 **프로그래밍 언어**뿐만 아니라, HTML이나 XML같은 언어를 처리하는 **브라우저 시스템**도 이에 해당

3. 기타 처리 시스템 사례 - 언어처리 시스템



- ◆ 다음은 언어처리 시스템의 하나인 **컴파일러**의 처리 사례

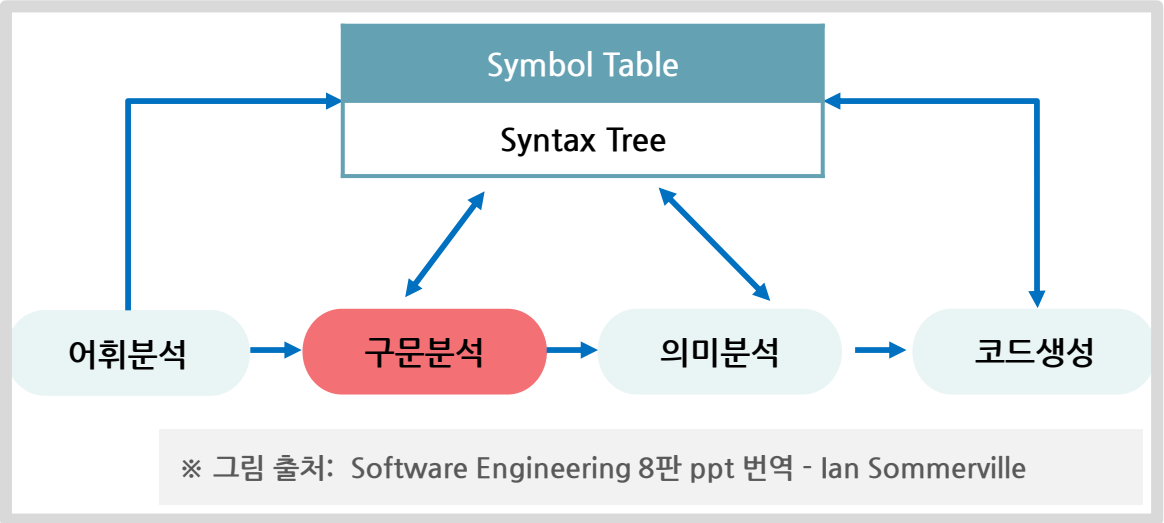


- ◆ 컴파일러는 맨 처음 코딩된 코드의 **어휘(낱개 단어)**를 먼저 분석(Lexical Analysis)함
- ◆ 분석 시 **심볼테이블(Symbol Table)**을 참고

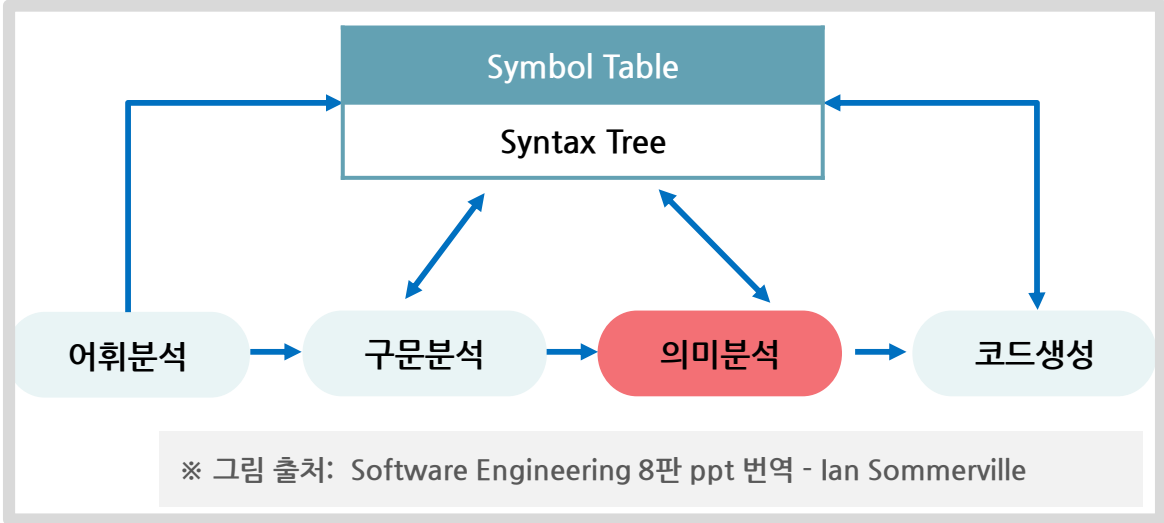
❖ 학습내용

[1] 응용시스템 아키텍처 참고모델

3. 기타 처리 시스템 사례 - 언어처리 시스템(계속)



◆ 어휘가 분석되면 **구문을(문법)** 분석(Syntactic Analysis)

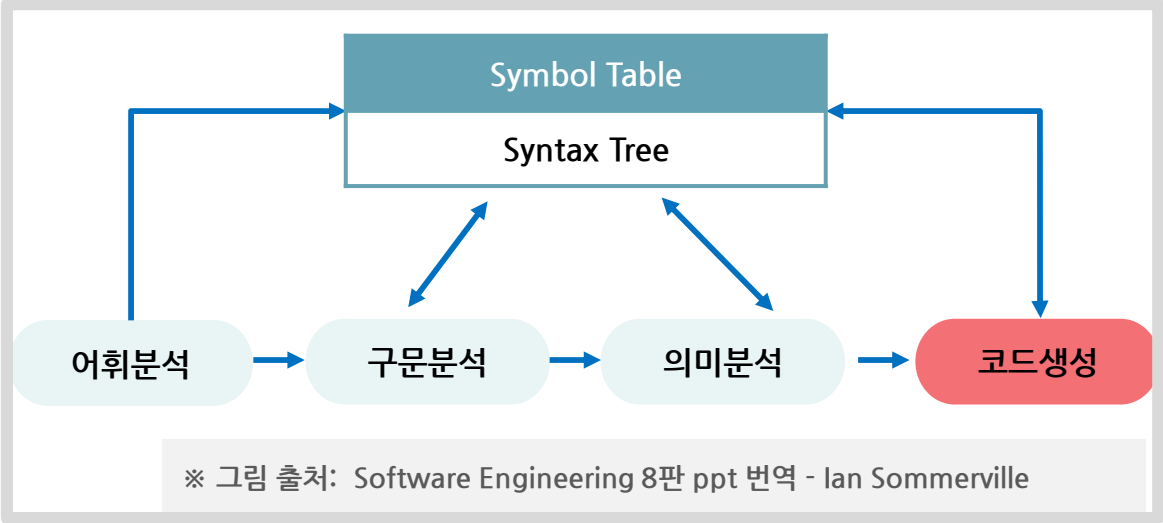


◆ 구문 분석 후 해당 구문을 묶어 최종적으로 어떤 일을 해야 되는지 **의미를 분석** (Semantic Analysis)

❖ 학습내용

[1] 응용시스템 아키텍처 참고모델

3. 기타 처리 시스템 사례 - 언어처리 시스템(계속)



◆ 마지막으로 해석된 코드를 **기계어 코드**로 생성(Code Generation)

3. 기타 시스템(계속)

- ◆ 분산 시스템 아키텍처
 - 여러 개의 **분산**된 서버, 프로세서에 의하여 처리되는 형태의 아키텍처
 - 분산 시스템 아키텍처에서 고려하여야 할 사항



❖ 학습내용

[1] 응용시스템 아키텍처 참고모델

3. 기타 시스템(계속)

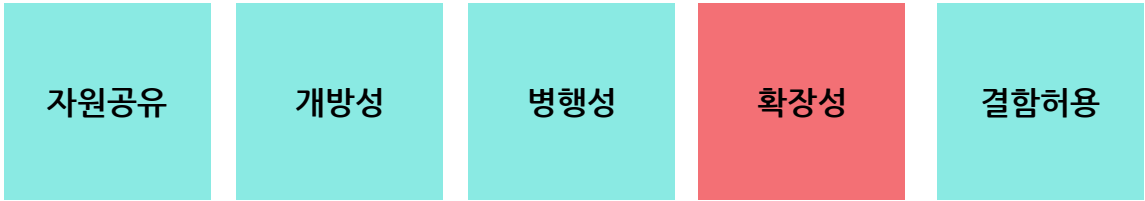
- ◆ 분산 시스템 아키텍처(계속)
 - 분산 시스템 아키텍처에서 고려하여야 할 사항



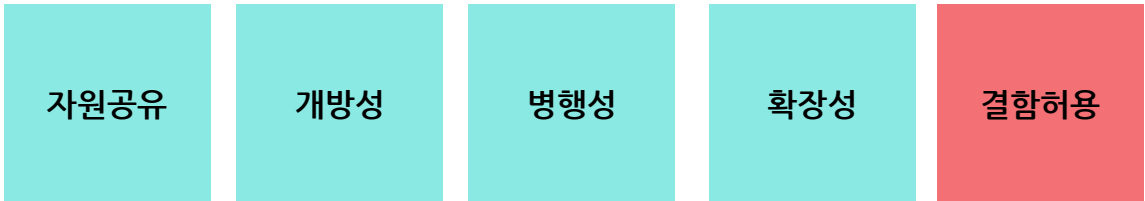
서로 다른 공급자의 장비와 소프트웨어들이 사용가능 하여야 함



성능증대를 위한 병행처리를 고려해야 함



새로운 리소스를 추가함으로써 처리능력이 향상되어야 함(Scalability)



장애가 발생하더라도 동작이 멈추지 않는 능력을 가져야 함(Fault Tolerance)

만일 설계하고자 하는 시스템이 분산 시스템 아키텍처라면 **자원공유, 개방성, 병행성, 확장성(Scalability), 결함허용(Fault Tolerance)**등을 고려하여야 함


❖ 학습내용

[2] 데이터베이스 설계

1. 데이터 설계 정의

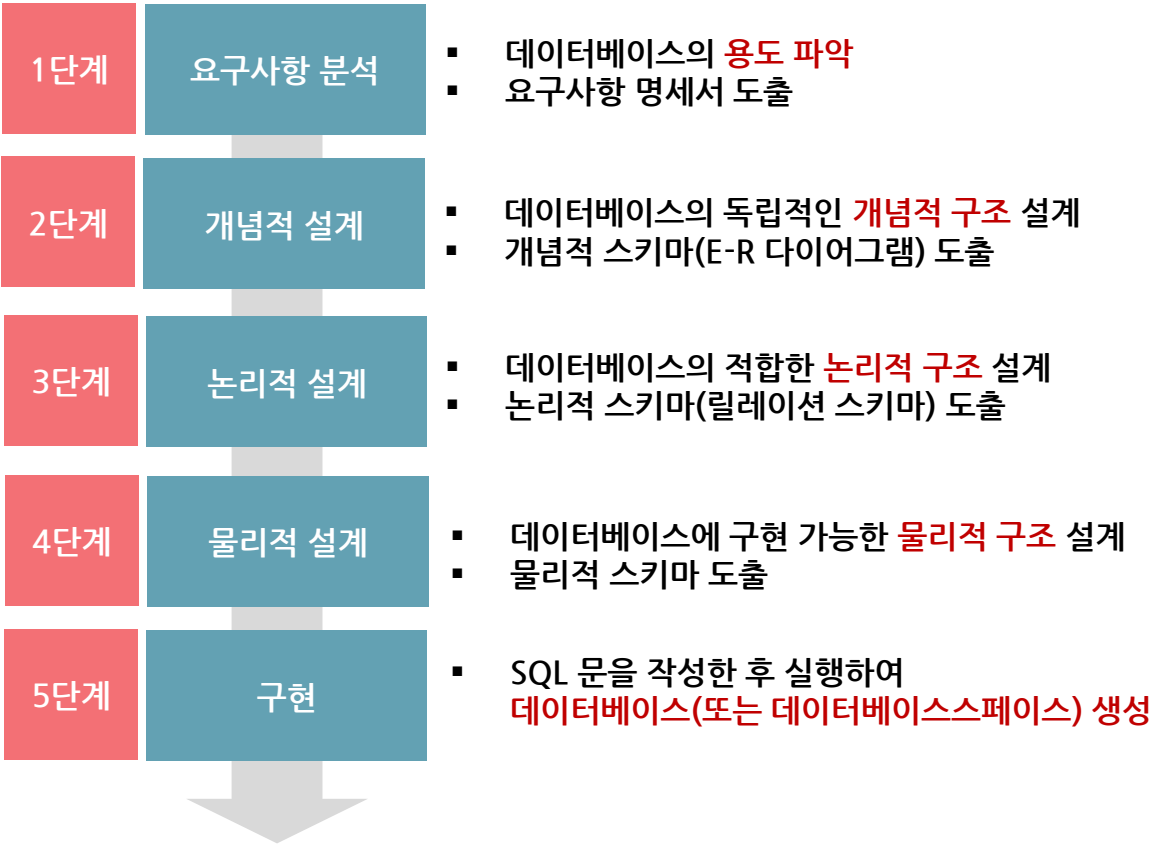
- ◆ 데이터 설계 시 고려사항
 - 설계단계에서 정보시스템에서 사용될 데이터, 즉 **데이터베이스를 설계**하는 것은 중요한 사항 중 하나임
- ◆ 데이터 설계 시 고려사항

 **중요한 데이터**는 무엇인가?

 데이터를 어떻게 **표현**할 것인가?

 데이터를 어떻게 **저장**할 것인가?

◆ 데이터 설계 진행 프로세스



❖ 학습내용

[2] 데이터베이스 설계

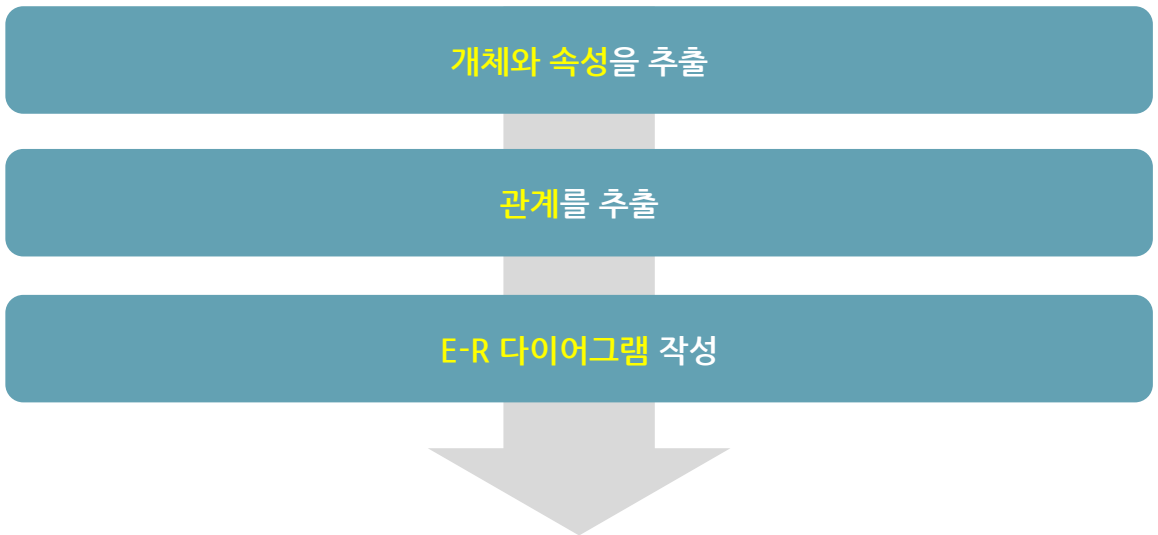
2. 개념적 데이터 설계

◆ 요구분석

- 사용자 요구사항 정의서를 바탕으로 **데이터와 관련된 부분**을 추출
- 데이터베이스에 대한 사용자의 다양한 요구 사항을 수집하고 분석하여, 개발할 데이터베이스의 용도를 파악해 **데이터 관련요구 사항 명세서**를 작성
- 데이터베이스를 사용할 **주요 사용자의 범위 결정**
- 해당 사용자가 조직에서 수행하는 **업무를 분석**
- 사용자가 요구하는 **데이터베이스의 용도**를 결정짓는 설계의 첫 단계가 요구사항 단계임

◆ 개념적 데이터 설계 정의

- 요구 사항 분석 단계의 결과물을 **개념적 데이터 모델**을 이용하여 표현
- 개념적 데이터 모델은 사용자들의 요구 사항들을 분석한 결과에서, 앞으로 데이터베이스에 저장해둘 필요가 있다고 판단되는 **데이터 요소를 추출**하고 **데이터 요소 간의 관계**를 파악하여 이를 **표현**하는 단계



❖ 학습내용

[2] 데이터베이스 설계

2. 개념적 데이터 설계 사례 - 개념적 데이터 설계

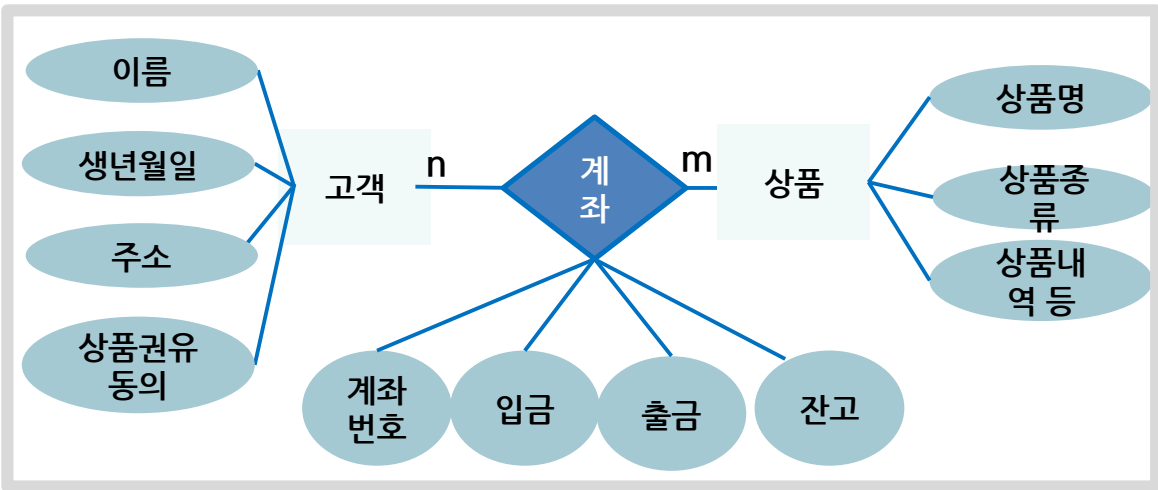
◆ 개체와 속성을 추출

개체	속성
↓	↓
고객	이름, 생년월일, 주소, 상품권유동의여부
계좌	계좌번호, 상품구분, 입금, 출금, 잔고
상품	상품명(독도사랑적금), 상품종류(예금, 적금, 대출)

◆ 관계를 추출

관계	관계에 참여하는 개체	관계유형	속성
↓	↓	↓	↓
계좌개설	고객, 계좌, 상품	다대다	계좌명, 상품종류, 계좌번호
입출금	계좌	일방	계좌번호, 입금, 출금, 잔고
고객가입 권유	고객, 상품	일대다	고객명, 상품권유동의여부, 상품명

◆ ERD의 작성



❖ 학습내용

[2] 데이터베이스 설계

3. 논리적, 물리적 데이터 설계

◆ 논리적 데이터 설계

- 개발에 사용할 DBMS에 적합한 논리적 데이터 모델을 이용해서, 개념적 설계 단계에서 생성한 **개념적 스키마**를 기반으로 **논리적 스키마**를 설계
- 즉, DBMS **독립적인 개념적 스키마**로부터 사용할 DBMS가 처리할 수 있는 데이터베이스의 **논리적 구조**를 설계하는 것이 논리적 설계 단계의 목표
- DBMS의 종류에 따라 네트워크 데이터 모델, 계층 데이터 모델, 관계 데이터 모델, 객체지향 데이터 모델 등 다양한 논리적 데이터 모델을 사용할 수 있지만 일반적으로 **관계 데이터 모델**을 많이 사용
- 논리적 설계 단계에서는 관계 데이터 모델을 이용하여 개념적 설계 단계의 결과물인 **E-R 다이어그램**을 관계 데이터 모델의 **릴레이션 스키마**, 즉 테이블 스키마로 변환하는 작업을 진행
- 5가지 **릴레이션 변환 규칙**에 따라 개념적 설계를 릴레이션 스키마(논리적 스키마)로 변환

1 모든 개체는 **릴레이션**으로 변환한다.

2 다대다($n:m$) 관계는 **릴레이션**으로 변환한다.

3 일대다($1:n$) 관계는 **외래키**로 표현한다.

4 일대일($1:1$) 관계는 **외래키**로 표현한다.

5 다중 값 속성은 독립 **릴레이션**으로 변환한다.

❖ 학습내용

[2] 데이터베이스 설계

3. 논리적, 물리적 데이터 설계(계속)

◆ 물리적 데이터 설계 및 구현

- 물리적 설계 단계에서는 하드웨어나 운영체제의 특성을 고려하여 필요한 인덱스의 구조나 내부 저장 구조 등에 대한 **물리적인 구조**를 설계
- 마지막으로 DBMS를 이용해 **SQL문**을 작성
- SQL문을 실행시켜서 데이터베이스를 실제로 생성하면 **데이터베이스 개발이 완료**

3. 논리적, 물리적 데이터 설계 사례 - 물리적 데이터 설계 SQL문

```
create table gongji(id int not null primary key auto_increment,  
                  title varchar (70),  
                  date date,  
                  content text);
```

❖ 학습내용

[3] 인터페이스 설계

1. I/O 인터페이스 설계

◆ 인터페이스 설계 개념

- 인터페이스의 사전적 의미는 **접점**의 개념
- 일반적인 정보시스템은 다수의 프로세스나 다수의 서버시스템간의 상호 정보의 교환, 입출력(I/O, Input, Output)을 주고 받기 때문에 **서로 상호 교환할 입출력의 설계**가 필요함
- 또한 정보시스템의 사용자가 존재한다면, 사용자가 정보시스템을 사용하기 위하여 단말기(PC, Web, 모바일 또는 기타 기기)의 **사용자가 접하는 정보화면**에 대한 설계가 필요함
- 즉 인터페이스 설계는 **I/O인터페이스 설계 및 사용자 인터페이스 설계**를 진행하며 **시스템의 성격**에 따라서 어느 하나의 인터페이스 설계가 더 중요할 수도 있음

예

- 회사소개 홈페이지 구축 프로젝트 ⇒ 화면에 대한 설계가 중요
- 은행상호간 자금이체 시스템 ⇒ IO인터페이스가 중요할 수 있음

◆ I/O인터페이스 설계

- 시스템, 모듈 간 주고받는 데이터(I/O정의), 함수, 클래스 등의 매개변수와 같은 인터페이스를 설계함

2. 사용자 인터페이스 설계

◆ 사용자 인터페이스 설계 개념

- 사용자 인터페이스 설계는 **사용자가 시스템의 접근하는 부분**을 설계하는 부분
- 일반적으로 **화면설계**가 대부분을 차지함

❖ 학습내용

[3] 인터페이스 설계

2. 사용자 인터페이스 설계(계속)

◆ 사용자 인터페이스 설계 개념



- 사용자 인터페이스는 해당시스템을 사용할 사용자가 **익숙**하고, 다른 시스템에서 경험한 것과 **유사**하고, 인터페이스에서 이것을 누르면 어떤 것이 실행될 것이라고 본인이 **기대된 기능**이 당연히 실행되도록 설계되어야 함
- 시스템 사용자는 시스템의 기능보다는 **사용자 인터페이스**로 시스템을 판단함
- 나쁜 인터페이스 설계는 시스템 마비상태 정도의 **에러**를 **유발**할 수도 있음
- 나쁜 사용자 인터페이스 설계는 수많은 소프트웨어 시스템을 절대 **사용하지 않게** 하는 원인이기도 함

◆ 사용자 인터페이스 설계 원칙

- 사용자 인터페이스 설계 시 다음의 원칙으로 설계

사용자 친밀성 (User Familiarity)	해당시스템을 가장 많이 사용하는 사람이 익숙 하도록 그려져야 함
일관성 (Consistency)	어디서나 가능하고, 비슷한 기능은 같은 방법으로 사용될 수 있는 일관성 을 유지 해야 함
최소의 당황 (Minimal Surprise)	시스템의 행동이 사용자를 놀라게 해서는 안 됨

❖ 학습내용

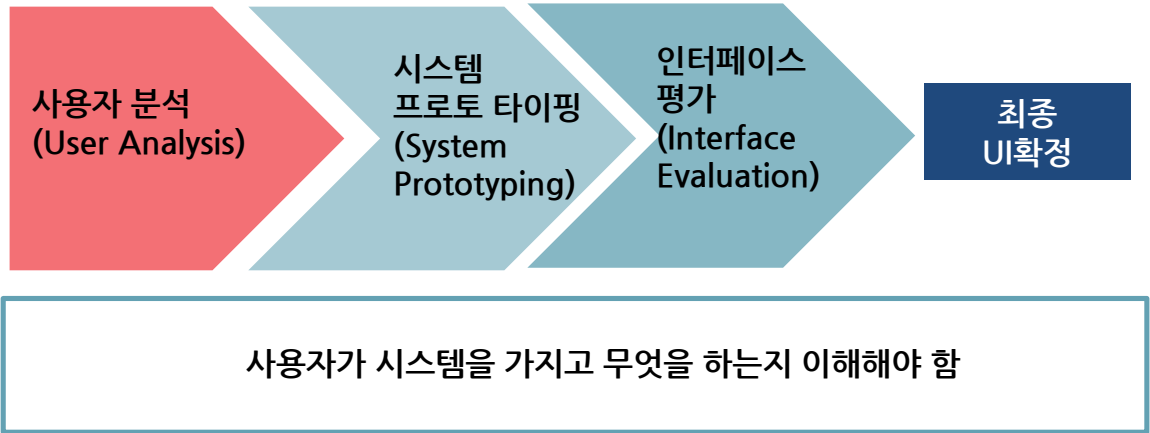
[3] 인터페이스 설계

2. 사용자 인터페이스 설계(계속)

- ◆ 사용자 인터페이스 설계 원칙(계속)
 - 사용자 인터페이스 설계 시 다음의 원칙으로 설계

복구가능성 (Recoverability)	에러(실수)로 인한 사용이 허용된 사용자의 복구기능이 포함되어야 함
사용자 안내 (User Guidance)	에러 발생 시 에러 의미에 대한 전달과 사용자 도움말 기능을 제공하여야 함
사용자 다양성 (User Diversity)	인터페이스는 다양한 시스템 사용자의 타입에 따른 접근 기능을 제공하여야 함

- ◆ 사용자 인터페이스(UI: User Interface) 설계 프로세스
 - UI설계 프로세스는 다음 3가지 주요 활동으로 구성

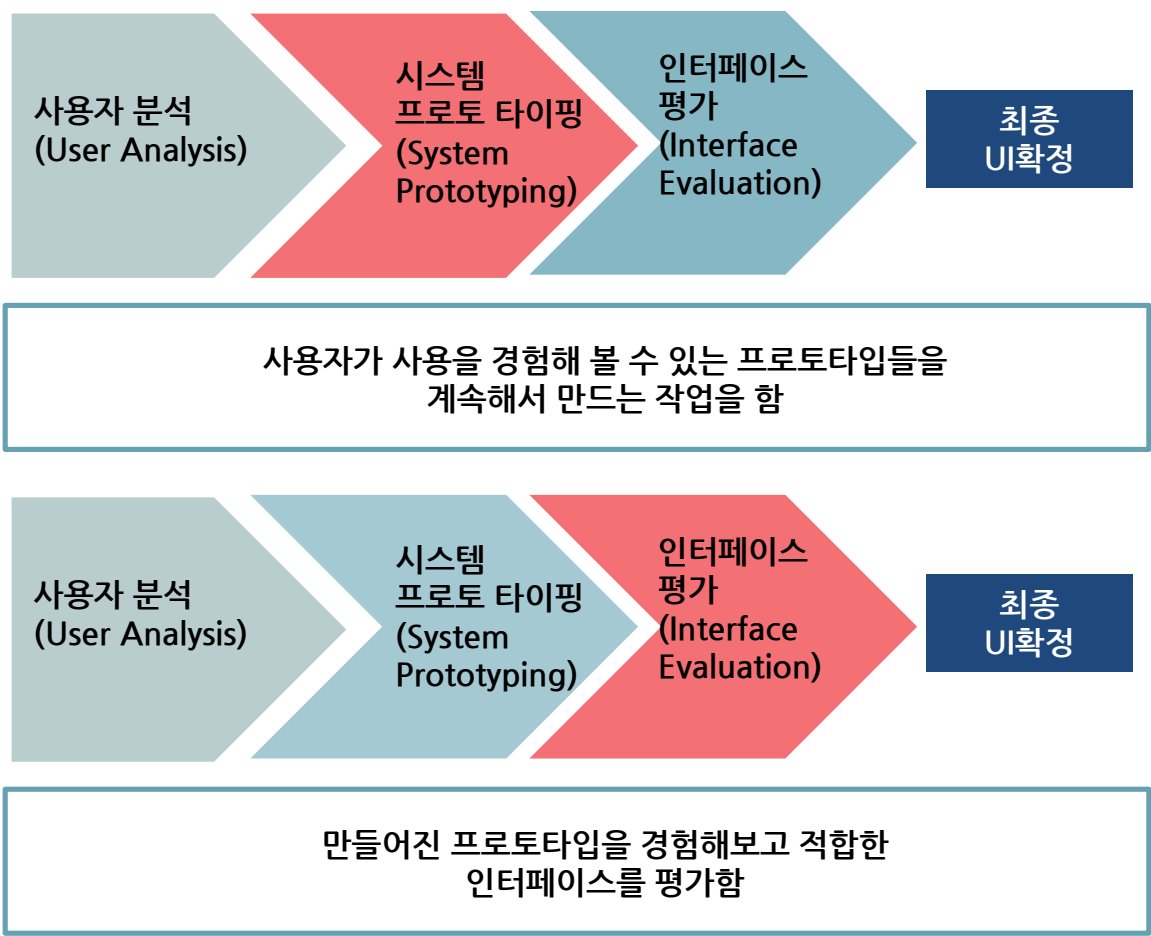


❖ 학습내용

[3] 인터페이스 설계

2. 사용자 인터페이스 설계(계속)

- ◆ 사용자 인터페이스(UI: User Interface) 설계 프로세스(계속)
 - UI설계 프로세스는 다음 3가지 주요 활동으로 구성(계속)



❖ 학습내용

[3] 인터페이스 설계

3. 정보 구조(IA) 설계

◆ 정보 구조의 정의

- 정보 구조(Information Architecture)
- 엔티티형이나 엔티티 집합으로 구성되는 **정보 모델링**으로 얻어지는 결과
- 각 엔티티 집합은 몇 개의 속성들로 표현되며, 각 **속성**은 현실 객체들이 가질 수 있는 값들의 이름
- 웹사이트 공간에서 **상호작용**을 설계하는 것이며 따라서 사용자의 경험을 이용하여 사용자의 두뇌에 인식하도록 만드는 것
- 실질적으로 웹사이트의 경우 **전체 Site Map**을 먼저 작성해보면 전체 웹사이트의 분석 설계 및 성격을 이해하는데 용이함
- 즉 화면 설계 작업에서 먼저 **어떠한 화면 메뉴체계**로 구성할 것인지 먼저 결정하여야 함

3. 정보 구조(IA) 설계 사례 - 정보 구조

- ◆ 어떠한 웹사이트의 **메뉴 구성**을 보면 해당 웹사이트의 어떠한 **정보**를 담고 있는 화면으로 구성되어 있는지, 더 나아가 해당 웹 시스템의 **성격과 체계**를 알 수 있음



※ 그림 출처: 한기대, 능력개발교육원

❖ 학습내용

[3] 인터페이스 설계

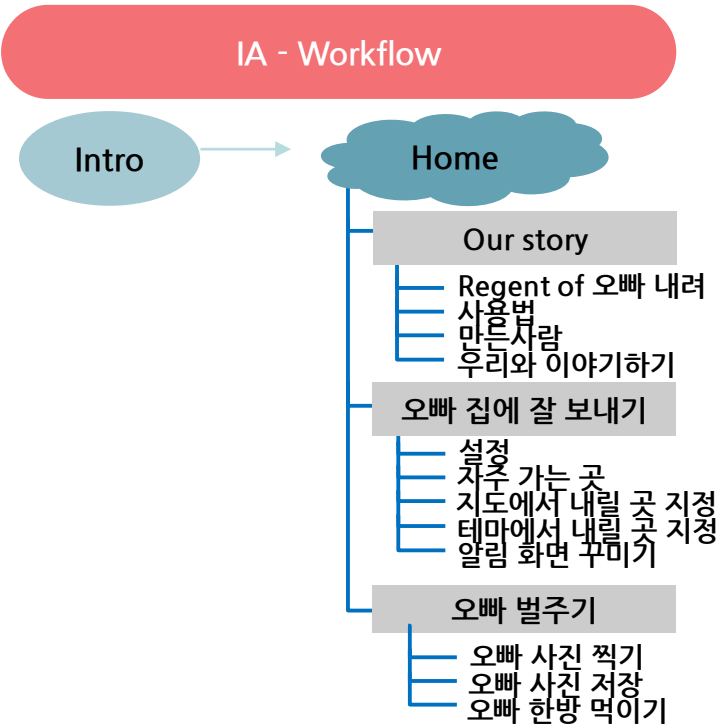
3. 정보 구조(IA) 설계(계속)

◆ 정보 구조 설계의 예

- 간략한 전체 시스템의 **스토리 보드**를 전개하고 이를 정보 구조로 구체화

3. 정보 구조(IA) 설계 사례 - 정보 구조 설계

◆ 원하는 곳에 도착하면 알림을 주는 위치기반 App을 설계한 사례



Now-trend: Story-telling

새벽 별을 보고 출근하여, 손톱 달을 보고 퇴근하는 IT가이 강철민은 무적의 솔로부대, 무적의 야근 부대, 무적의 똥배부대원이다. 그의 기지는 경기 남부 모처의 신도시에 위치한다. 매번 막차 퇴근 버스의 지친 몸을 기고 자리에 앉아서 퇴근 투어를 할 때 마다, 폭포수의 침의 한 바가지 쏟아내며 골아 떨어진다. 그때 들리는 산신령의 목소리 ~~이봐 청춘, 종점이야 얼렁 내려~~

오늘 용돈은 점심값과 KT&G에 납부했다.
종점에서 우리 집까지 택시비도 없다.
아 나도 버스가 집에 도착할 즈음, 나를 깨워주는 소녀가 있었으면...

❖ 핵심정리

1. 응용시스템 아키텍처 참고모델

- 어플리케이션(응용, 소프트웨어) 관점에서 아키텍처를 분류하면 데이터 처리 시스템, 트랜잭션 처리 시스템, 이벤트 처리 시스템, 언어처리 시스템 등의 모델이 많이 사용됨
- 데이터처리 응용 시스템은 데이터의 처리 관리를 중점으로 다루는 시스템
- 정보에 대한 요청에 대하여 처리하고, 정보를 갱신 하는 등 실시간 트랜잭션 처리 중심 응용시스템

2. 데이터베이스 설계

- 데이터 설계는 요구분석, 개념적 데이터 설계, 논리적 데이터 설계, 물리적 데이터 설계, 구현 순으로 진행

3. 인터페이스 설계

- 일반적인 정보시스템은 다수의 프로세스나 다수의 서버시스템간의 상호 정보의 교환, 입출력(I/O, Input, Output)을 주고 받기 때문에 서로 상호 교환할 입출력의 설계가 필요함
- 정보시스템의 사용자가 존재한다면, 사용자가 정보시스템을 사용하기 위하여 단말기(PC, Web, 모바일 또는 기타 기기)의 사용자가 접하는 정보화면에 대한 설계가 필요함