

대분류/20  
정보통신

중분류/01  
정보기술

소분류/02  
정보기술개발

세분류/02  
응용SW엔지니어링

능력단위/32

NCS학습모듈

# 응용SW 기초 기술 활용

LM2001020232\_19v4



교육부

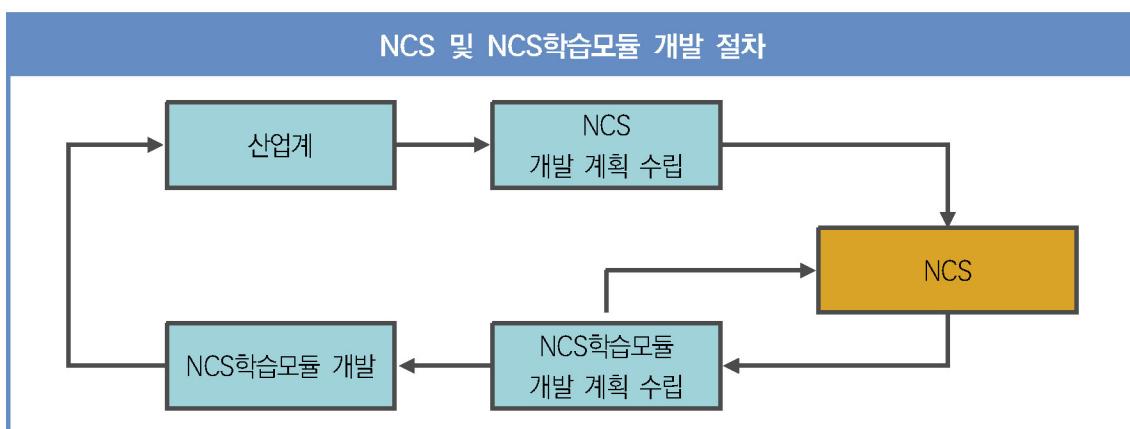
NCS 학습모듈은 교육훈련기관에서 출처를 명시하고 교육적 목적으로 활용할 수 있습니다. 다만 NCS 학습모듈에는 국가(교육부)가 저작재산권 일체를 보유하지 않은 저작물들(출처가 표기되어 있는 도표, 사진, 삽화, 도면 등)이 포함되어 있으므로 이러한 저작물들의 변형, 복제, 공연, 배포, 공중 송신 등과 이러한 저작물을 활용한 2차 저작물의 생성을 위해서는 반드시 원작자의 동의를 받아야 합니다.

# NCS학습모듈의 이해

※ 본 NCS학습모듈은 「NCS 국가직무능력표준」사이트(<http://www.ncs.go.kr>)에서 확인 및 다운로드할 수 있습니다.

## I NCS학습모듈이란?

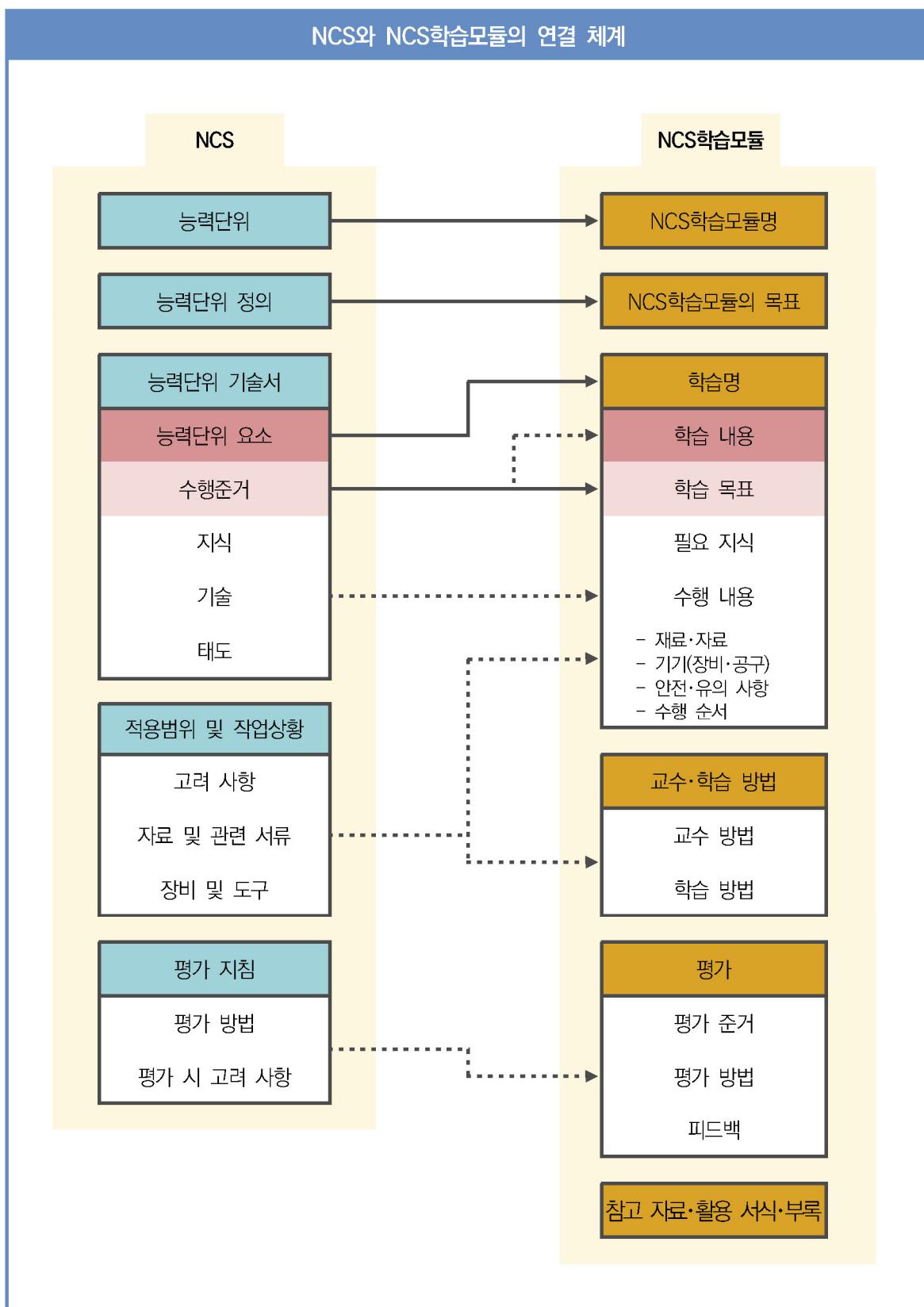
- 국가직무능력표준(NCS: National Competency Standards)이란 산업현장에서 직무를 수행하기 위해 요구되는 지식·기술·소양 등의 내용을 국가가 산업부문별·수준별로 체계화한 것으로 산업현장의 직무를 성공적으로 수행하기 위해 필요한 능력(지식, 기술, 태도)을 국가적 차원에서 표준화한 것을 의미합니다.
- 국가직무능력표준(이하 NCS)이 현장의 ‘직무 요구서’라고 한다면, **NCS학습모듈은 NCS의 능력단위를 교육훈련에서 학습할 수 있도록 구성한 ‘교수·학습 자료’입니다.** NCS학습모듈은 구체적 직무를 학습할 수 있도록 이론 및 실습과 관련된 내용을 상세하게 제시하고 있습니다.



- **NCS학습모듈은 다음과 같은 특징을 가지고 있습니다.**

- 첫째, NCS학습모듈은 산업계에서 요구하는 직무능력을 교육훈련 현장에 활용할 수 있도록 성취목표와 학습의 방향을 명확히 제시하는 가이드라인의 역할을 합니다.
- 둘째, NCS학습모듈은 특성화고, 마이스터고, 전문대학, 4년제 대학교의 교육기관 및 훈련기관, 직장교육기관 등에서 표준교재로 활용할 수 있으며 교육과정 개편 시에도 유용하게 참고할 수 있습니다.

○ NCS와 NCS학습모듈 간의 연결 체계를 살펴보면 아래 그림과 같습니다.



## II NCS학습모듈의 체계

- NCS학습모듈은 1. NCS학습모듈의 위치 , 2. NCS학습모듈의 개요 , 3. NCS학습모듈의 내용 체계 , 4. 참고 자료 , 5. 활용서식/부록 으로 구성되어 있습니다.

### 1. NCS학습모듈의 위치

- NCS학습모듈의 위치는 NCS 분류 체계에서 해당 학습모듈이 어디에 위치하는지를 한 눈에 볼 수 있도록 그림으로 제시한 것입니다.

[NCS-학습모듈의 위치]		
대분류	문화·예술·디자인·방송	
중분류	문화콘텐츠	
소분류	문화콘텐츠제작	
세분류		
방송콘텐츠제작	능력단위	학습모듈명
영화콘텐츠제작	프로그램 기획	프로그램 기획
음악콘텐츠제작	아이템 선정	아이템 선정
광고콘텐츠제작	자료 조사	자료 조사
게임콘텐츠제작	프로그램 구성	프로그램 구성
애니메이션 콘텐츠제작	캐스팅	캐스팅
만화콘텐츠제작	제작계획	제작계획
캐릭터제작	방송 미술 준비	방송 미술 준비
스마트문화앱 콘텐츠제작	방송 리허설	방송 리허설
영사	야외촬영	야외촬영
	스튜디오 제작	스튜디오 제작
	...	...

**학습모듈은**

NCS 능력단위 1개당 1개의 학습모듈 개발을 원칙으로 합니다. 그러나 필요에 따라 고용단위 및 교과단위를 고려하여 능력단위 몇 개를 묶어 1개 학습모듈로 개발할 수 있으며, NCS 능력단위 1개를 여러 개의 학습모듈로 나누어 개발할 수도 있습니다.

## 2. NCS학습모듈의 개요

○ NCS학습모듈의 개요는 학습모듈이 포함하고 있는 내용을 개략적으로 설명한 것으로

학습모듈의 목표, 선수학습, 학습모듈의 내용 체계, 핵심 용어로 구성되어 있습니다.

### 학습모듈의 목표

해당 NCS 능력단위의 정의를 토대로 학습 목표를 작성한 것입니다.

### 선수학습

해당 학습모듈에 대한 효과적인 교수·학습을 위하여 사전에 이수해야 하는 학습모듈, 학습 내용, 관련 교과목 등을 기술한 것입니다.

### 학습모듈의 내용 체계

해당 NCS 능력단위요소가 학습모듈에서 구조화된 체계를 제시한 것입니다.

### 핵심 용어

해당 학습모듈의 학습 내용, 수행 내용, 설비·기자재 등 가운데 핵심적인 용어를 제시한 것입니다.

## 제작계획 학습모듈의 개요

### 학습모듈의 목표

본격적인 촬영을 준비하는 단계로서, 촬영 대본을 확정하고 제작 스태프를 조직하며 촬영 장비와 촬영 소품을 준비할 수 있다.

### 선수학습

제작 준비(LM0803020105\_13v1), 섭외 및 제작스태프 구성(LM0803020104\_13v1), 촬영 제작(LM0803020106\_13v1), 촬영 장비 준비(LM0803040204\_13v1.4), 미술 디자인 협의하기(LM0803040203\_13v1.4)

### 학습모듈의 내용체계

학습	학습 내용	NCS 능력단위 요소	
		코드번호	요소 명칭
1. 촬영 대본 확정하기	1-1. 촬영 구성안 검토와 수정	0803020114_16.3.1	촬영 대본 확정하기
2. 제작 스태프 조직하기	2-1. 기술 스태프 조직 2-2. 미술 스태프 조직 2-3. 전문 스태프 조직	0803020114_16.3.2	제작 스태프 조직하기
3. 촬영 장비 계획하기	3-1. 촬영 장비 점검과 준비	0803020114_16.3.3	촬영 장비 계획하기
4. 촬영 소품 계획하기	4-1. 촬영 소품 목록 작성 4-2. 촬영 소품 제작 의뢰	0803020114_16.3.4	촬영 소품 계획하기

### 핵심 용어

촬영 구성안, 제작 스태프, 촬영 장비, 촬영 소품

### 학습모듈의 목표는

학습자가 해당 학습모듈을 통해 성취해야 할 목표를 제시한 것으로, 교수자는 학습자가 학습모듈의 전체적인 내용흐름을 파악하도록 지도할 수 있습니다.

### 선수학습은

교수자 또는 학습자가 해당 학습모듈을 교수·학습하기 이전에 이수해야 하는 교과목 또는 학습모듈(NCS 능력단위) 등을 표기한 것입니다. 따라서 교수자는 학습자가 개별 학습, 자기 주도 학습, 방과 후 활동 등 다양한 방법을 통해 이수할 수 있도록 지도하는 것을 권장합니다.

### 핵심 용어는

해당 학습모듈을 대표하는 주요 용어입니다. 학습자가 해당 학습모듈을 통해 학습하고 평가받게 될 주요 내용을 알 수 있습니다. 「NCS 국가직무능력표준」 사이트 ([www.ncs.go.kr](http://www.ncs.go.kr))의 색인 (찾아보기) 중 하나로 이용할 수 있습니다.

### 3. NCS학습모듈의 내용 체계

○ NCS학습모듈의 내용은 크게 **학습**, **학습 내용**, **교수·학습 방법**, **평가**로 구성되어 있습니다.

학습	해당 NCS 능력단위요소 명칭을 사용하여 제시한 것입니다. 학습은 크게 학습 내용, 교수·학습 방법, 평가로 구성되며 해당 NCS 능력단위의 능력단위 요소별 지식, 기술, 태도 등을 토대로 내용을 제시한 것입니다.
학습 내용	학습 내용은 학습 목표, 필요 지식, 수행 내용으로 구성되며, 수행 내용은 재료·자료, 기기(장비·공구), 안전·유의 사항, 수행 순서, 수행 tip으로 구성한 것입니다. 학습모듈의 학습 내용은 실제 산업현장에서 이루어지는 업무활동을 표준화된 프로세스에 기반하여 다양한 방식으로 반영한 것입니다.
교수·학습 방법	학습 목표를 성취하기 위한 교수자와 학습자 간, 학습자와 학습자 간 상호 작용이 활발하게 일어날 수 있도록 교수자의 활동 및 교수 전략, 학습자의 활동을 제시한 것입니다.
평가	평가는 해당 학습모듈의 학습 정도를 확인할 수 있는 평가 준거 및 평가 방법, 평가 결과의 피드백 방법을 제시한 것입니다.

학습 1	촬영 대본 확정하기
학습 2	<b>제작 스태프 조직하기</b>
학습 3	촬영 장비 계획하기
학습 4	촬영 소품 계획하기

**2-1. 기술 스태프 조직**

**학습 목표** • 프로그램 제작에 적합한 기술 스태프를 조직할 수 있다.

**필요 지식** /

① 기술 스태프의 구성  
프로그램의 장르에 따라 구성하는 기술 스태프는 많은 차이가 있다. 같은 장르의 프로그램이라도 그 형식이나 내용, 규모에 따라서 구성되는 기술 스태프의 종류와 인원 수는 천차만별이다.

1. 스튜디오 프로그램  
토크쇼, 종합 구성, 예능과 같은 스튜디오 프로그램은 부조정실과 스튜디오를 사용하여 제작하기 때문에 많은 기술 스태프가 필요하다.

**학습은**  
해당 NCS 능력단위요소 명칭을 사용하여 제시하였습니다. 하나의 학습은 일반교과의 ‘대단원’에 해당되며, 학습모듈을 구성하는 가장 큰 단위가 됩니다. 또한 하나의 직무를 수행하기 위한 가장 기본적인 단위로 사용할 수 있습니다

**학습 내용은**  
NCS 능력단위요소별 수행준거를 기준으로 제시하였습니다. 일반교과의 ‘중단원’에 해당합니다.

**학습 목표는**  
학습 내용을 이수할 때 학습자가 갖춰야 할 행동 수준을 의미합니다. 따라서 수업시간의 과목 목표로 활용할 수 있습니다.

**필요 지식은**  
해당 NCS의 지식을 토대로 학습에 대한 이해와 성과를 제고하기 위해 반드시 알아야 할 주요 지식을 제시하였습니다. 필요 지식은 수행에 꼭 필요한 핵심 내용을 위주로 제시하여 교수자의 역할이 매우 중요하며, 이후 수행 순서와 연계하여 교수·학습으로 진행할 수 있습니다.

수행 내용	/ 기술 스태프 구성표 작성하기	
<p><b>재료·자료</b></p> <ul style="list-style-type: none"> <li>방송프로그램 제작 기획서 및 방송 대본, 콘티(continuity), 제작 일정, 운용표</li> <li>장비 및 시설, 제작 시설 배정 의뢰서 및 배정표, 방송 기술 스태프 데이터베이스(DB) 자료</li> </ul> <p><b>기기(장비·공구)</b></p> <ul style="list-style-type: none"> <li>컴퓨터 등</li> </ul> <p><b>안전·유의 사항</b></p> <ul style="list-style-type: none"> <li>프로그램의 내용과 제작 방법을 분석하고, 각 스태프들의 역할을 신중하게 검토한다.</li> </ul> <p><b>수행 순서</b></p> <ol style="list-style-type: none"> <li>방송 대본이나 콘티(continuity), 큐 시트를 분석하고, 프로그램의 내용적 특성, 제작 과정에 대한 자료를 수집한다.</li> <li>프로그램 제작 방법을 결정한다.             <ul style="list-style-type: none"> <li>스튜디오 녹화를 할 것인가, 야외 촬영을 할 것인가 검토한다.</li> </ul> </li> </ol> <p><b>수행 tip</b></p> <ul style="list-style-type: none"> <li>스태프의 결정은 스태프 간의 호흡을 중요시하여 선정해야 프로그램의 질을 향상시킬 수 있다.</li> </ul>		<p><b>수행 내용은</b></p> <p>해당 학습모듈에서 제시한 내용 중 기술(skill)을 습득하기 위한 실습과제로 활용할 수 있습니다.</p> <p><b>재료·자료는</b></p> <p>수행 내용을 수행하는데 필요한 재료 및 준비물로 실습 시 활용할 수 있습니다.</p> <p><b>기기(장비·공구)는</b></p> <p>수행 내용에 필요한 기본적인 장비 및 도구를 제시하였습니다. 제시된 기기 외에도 수행에 필요한 다양한 도구나 장비를 활용할 수 있습니다.</p> <p><b>안전·유의사항은</b></p> <p>수행 내용을 수행하는 데 있어 안전상 주의해야 할 점 및 유의사항을 제시하였습니다. 실습 시 유념해야 하며, NCS의 고려사항도 추가적으로 활용할 수 있습니다.</p> <p><b>수행 순서는</b></p> <p>실습 과제의 진행 순서로 활용할 수 있습니다.</p> <p><b>수행 tip은</b></p> <p>수행 내용에서 실습을 용이하게 할 수 있는 아이디어를 제시하였습니다. 수행 tip은 지도상의 안전 및 유의사항 외에 전반적으로 적용되는 주인점 및 수행 과제 목적에 대한 보충설명, 추가사항 등으로 활용할 수 있습니다.</p>
<p><b>학습2</b></p> <p><b>교수·학습 방법</b></p> <p><b>교수 방법</b></p> <ul style="list-style-type: none"> <li>방송 프로그램의 기술적 요소, 미술 구성 요소, 특수 촬영에 대해서 설명한다.</li> <li>방송 프로그램 제작에서 각 기술 스태프의 역할에 대해 설명한다.</li> <li>방송 프로그램을 분석하고 필요한 기술 스태프를 구성할 수 있도록 지시한다.</li> </ul> <p><b>학습 방법</b></p> <ul style="list-style-type: none"> <li>방송 프로그램의 기술적 요소, 미술 구성 요소, 특수 촬영에 대해서 알아본다.</li> <li>프로그램 제작에 필요한 기술 스태프의 역할을 이해하고, 기술 스태프 구성표를 작성한다.</li> </ul>		<p><b>교수·학습 방법은</b></p> <p>학습 목표를 성취하는 데 필요한 교수 방법과 학습 방법을 제시하였습니다.</p> <p><b>교수 방법은</b></p> <p>해당 학습 활동에 필요한 학습 내용, 학습 내용과 관련된 자료명, 자료 형태, 수행 내용의 진행 방식 등에 대하여 제시하였습니다. 또한 학습자의 수업참여도 제고 방법 및 수업 진행상 유의사항 등도 제시하였습니다. 선수학습이 필요한 학습을 학습자가 숙지하였는지 교수자가 확인하는 과정으로 활용할 수도 있습니다.</p> <p><b>학습 방법은</b></p> <p>해당 학습 활동에 필요한 학습자의 자기 주도 학습 방법을 제시하였습니다. 또한 학습자가 숙달해야 할 실기 능력과 학습 과정에서 주의해야 할 사항 등도 제시하였습니다. 학습자가 학습을 이수하기 전 반드시 숙지해야 할 기본 지식을 학습하였는지 스스로 확인하는 과정에 활용할 수 있습니다.</p>

## 학습2 평가

### 평가 준거

- 평가자는 학습자가 학습 목표를 성공적으로 달성하였는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.

학습 내용	학습 목표	성취수준 상 중 하
기술 스태프 조직	- 프로그램 제작에 적합한 기술 스태프를 조직할 수 있다.	
미술 스태프 조직	- 프로그램 제작에 적합한 미술 스태프를 조직할 수 있다.	
전문 스태프 조직	- 프로그램 특수 촬영을 위한 전문 스태프를 조직할 수 있다.	

### 평가 방법

- 사례 연구

학습 내용	평가 항목	성취수준 상 중 하
기술 스태프 조직	- 프로그램에서 기술적 요소의 파악 여부 - 기술 스태프의 역할 파악 여부 - 프로그램에 필요한 기술 스태프 구성표 작성 능력	

### 피드백

- 사례 연구
  - 프로그램을 선택하여 기술 스태프, 미술 스태프, 전문 스태프 구성표를 예시와 같이 작성하였는지 개인별 능력을 평가한 후, 그 결과를 모든 학습자에게 공유하도록 한다.

### 평가는

NCS 능력단위의 평가 방법과 평가 시 고려사항을 종용하여 작성합니다. 교수자와 학습자가 평가 항목별 성취수준 확인 시 활용할 수 있습니다.

### 평가 준거는

학습자가 학습을 어느 정도 성취하였는지 평가하기 위한 기준을 제시하고 있습니다. 학습 목표와 연계하여 단위수업 시간에 평가 항목 별 성취수준을 평가하는 데 활용할 수 있습니다.

### 평가 방법은

NCS 능력단위의 평가 방법을 참고하였으며, 평가 준거에 따른 평가 방법을 2개 이상 제시합니다. 평가 방법의 종류는 포트폴리오, 문제해결 시나리오, 서술형 시험, 논술형 시험, 사례 연구, 평가자 체크리스트, 작업장 평가 등이 있으며, NCS 능력단위 요소 별 수행 수준을 평가하는 데 가장 적절한 방법을 선정하여 활용할 수 있습니다.

### 피드백은

평가 후에 학습자들에게 평가 결과를 피드백하여 학습 목표를 달성하는 데 활용할 수 있습니다.

## 4. 참고 자료

### 참고 자료

- 교육부(2013). 섭외 및 제작스태프 구성(LM0803020104\_13v1). 한국직업능력개발원.

### 참고 자료는

해당 학습모듈에 제시된 인용 자료의 출처를 제시하였습니다. 교수·학습의 과정에서 참고로 활용할 수 있습니다.

## 5. 활용 서식/부록

### 활용 서식

#### 스튜디오 기술 스태프 구성표

직종	이름	연락처	소속	특이사항	비고
기술감독					
조명감독					

### 활용 서식은

평가 서식, 실습 시트 등 교수·학습 시 활용할 수 있는 다양한 서식들로 구성하였습니다. 수행에서 평가에 이르기까지 필요한 서식을 해당 모듈의 특성에 맞춰 개발하거나 기존의 양식을 활용하여 제시하였습니다.

### 부록

#### [디지털 텔레비전 방송프로그램 음량 등에 관한 기준]

제정 2014. 11. 29. 미래창조과학부 고시 제2014-87호

#### 제1장 총칙

제1조(목적) 이 고시는 방송법 제70조의2제1항에 따라 방송사업자가 디지털 텔레비전 방송프로그램 및 방송광고의 음량을 일정하게 유지하기 위해 필요한 사항을 규정함을 목적으로 한다.

### 부록은

활용 서식 이외에 교수·학습 과정에서 참고할 수 있는 자료가 있는 경우 제시하였습니다.

## [NCS-학습모듈의 위치]

대분류	정보통신
중분류	정보기술
소분류	정보기술개발

세분류	능력단위	학습모듈명
SW아키텍처	요구사항 확인	요구사항 확인
응용SW엔지니어링	데이터 입출력 구현	데이터 입출력 구현
임베디드SW엔지니어링	통합 구현	통합 구현
DB엔지니어링	정보시스템 이행	정보시스템 이행
NW엔지니어링	제품소프트웨어 패키징	제품소프트웨어 패키징
보안엔지니어링	서버프로그램 구현	서버프로그램 구현
UI/UX엔지니어링	인터페이스 구현	인터페이스 구현
시스템SW엔지니어링	애플리케이션 배포	애플리케이션 배포
빅데이터플랫폼 구축	애플리케이션 리팩토링	애플리케이션 리팩토링
핀테크엔지니어링	인터페이스 설계	인터페이스 설계
데이터아키텍처	애플리케이션 요구사항 분석	애플리케이션 요구사항 분석
IoT시스템연동	기능 모델링	기능 모델링
인프라스트럭쳐 아키텍처 구축	애플리케이션 설계	애플리케이션 설계

정적모델 설계	정적모델 설계
동적모델 설계	동적모델 설계
화면 설계	화면 설계
화면 구현	화면 구현
애플리케이션 테스트 관리	애플리케이션 테스트 관리
애플리케이션 테스트 수행	애플리케이션 테스트 수행
소프트웨어공학 활용	소프트웨어공학 활용
소프트웨어개발 방법론 활용	소프트웨어개발 방법론 활용
프로그래밍 언어 응용	프로그래밍 언어 응용
프로그래밍 언어 활용	프로그래밍 언어 활용
<b>응용SW 기초 기술 활용</b>	<b>응용SW 기초 기술 활용</b>
개발자 환경 구축	개발자 환경 구축
개발 환경 운영 지원	개발 환경 운영 지원



---

# 차 례

---

학습모듈의 개요	1
학습 1. 네트워크 기초 활용하기	
1-1. 네트워크 프로토콜 활용	3
• 교수 · 학습 방법	22
• 평가	23
학습 2. 미들웨어 기초 활용하기	
2-1. 미들웨어 파악	25
2-2. 미들웨어 운용	34
• 교수 · 학습 방법	48
• 평가	49
학습 3. 데이터베이스 기초 활용하기	
3-1. 데이터베이스 특징 식별	51
3-2. 관계형 데이터베이스 테이블 정의	63
3-3. 관계형 데이터베이스 테이블 조작	75
• 교수 · 학습 방법	83
• 평가	84
참고 자료	86



# 응용SW 기초 기술 활용 학습모듈의 개요

## 학습모듈의 목표

응용소프트웨어 개발을 위하여 네트워크, 미들웨어, 데이터베이스의 기초 기술을 적용할 수 있다.

## 선수학습

개발자 환경 구축(2001020233\_19v4), 개발자 환경 운영 지원(2001020234\_19v1), 프로토콜 패킷 분석(2001020518\_19v4)

## 학습모듈의 내용체계

학습	학습 내용	NCS 능력단위 요소	
		코드번호	요소 명칭
1. 네트워크 기초 활용하기	1-1. 네트워크 프로토콜 활용	2001020232_19v4.1	네트워크 기초 활용하기
2. 미들웨어 기초 활용하기	2-1. 미들웨어 파악 2-2. 미들웨어 운영	2001020232_19v4.2	미들웨어 기초 활용하기
3. 데이터베이스 기초 활용하기	3-1. 데이터베이스 특징 식별 3-2. 관계형 데이터베이스 테이블 정의 3-3. 관계형 데이터베이스 테이블 조작	2001020232_19v4.3	데이터베이스 기초 활용하기

## 핵심 용어

네트워크 계층 구조, 프로토콜, TCP/UDP, 미들웨어, MVC, DBMS, ERD, SQL, CRUD, 정규화



## 학습 1

## 네트워크 기초 활용하기

학습 2

미들웨어 기초 활용하기

학습 3

데이터베이스 기초 활용하기

### 1-1. 네트워크 프로토콜 활용

#### 학습 목표

- 네트워크 계층 구조에서 각 계층의 역할을 구별할 수 있다.
- 패킷 스위칭 시스템의 라우팅 알고리즘과 프로토콜을 식별할 수 있다.
- 응용소프트웨어의 특성에 따라 프로토콜을 구별하여 적용할 수 있다.

#### 필요 지식

/

##### ① 네트워크 계층 구조

###### 1. OSI 7 계층(Open System Interconnection 7 Layer)

###### (1) OSI 7 계층의 개념

국제표준기구(ISO: International Organization for Standardization)에서 정리한 네트워크 구조에 대한 기본 모델이다.

###### (2) OSI 7 계층의 필요성

데이터 통신을 위해서는 데이터를 전기 신호로 변환하고 복잡한 네트워크 경로를 통해 데이터를 전달해야 하며 송신지와 수신지의 컴퓨터 구조가 상이할 경우 전송한 데이터가 올바르게 인식되지 않을 수도 있어 복잡한 변환 과정을 거쳐야 한다. 이런 복잡한 구성을 유사한 기능별로 계층화하여 처리되는 정보들을 캡슐화하여 구성하고 각 계층에 사용되는 통신 규격을 프로토콜로 표준화함으로써 응용소프트웨어 개발자, 서버/네트워크 엔지니어들이 본인이 연관된 계층만 고려하며 업무를 수행할 수 있도록 한다.

#### OSI 7 Layer의 응용소프트웨어 개발 활용

응용소프트웨어 개발자는 주로 OSI 7 Layer의 상위 계층(5~7)을 다루고 서버/네트워크 엔지니어는 하위 계층(1~4)을 다룬다. 응용소프트웨어 개발자는 4계층에 지정된 포트 번호를 통해 상위 계층의 프로토콜을 구별할 수 있기 때문에 응용소프트웨어 개발을 위해서는 4계층의 헤더 정보 분석에도 관심을 가지고 학습하도록 한다.



출처: 집필진 제작(2021)  
[그림 1-1] OSI 7 Layer

### (3) OSI 7 계층의 구성

데이터를 목적지에 정확히 전달하는 것에 초점을 둔 하위 계층과 송수신 데이터의 가공 및 활용에 초점을 둔 상위 계층으로 구성된다.

〈표 1-1〉 OSI 7 Layer 구성

구분	계층	역할 및 기능	주요 장비
Application Layer (상위 계층)	7. Application Layer	<ul style="list-style-type: none"> <li>- 사용자에 서비스 제공</li> <li>- 사용자 입출력 정의</li> <li>- 응용 프로세스 관리</li> </ul>	L7 스위치, 웹 방화벽
	6. Presentation Layer	<ul style="list-style-type: none"> <li>- 송수신지의 다른 데이터 표현 방식을 상호 인식 가능하도록 변환</li> <li>- 인코딩, 암호화, 압축, 코드 변환</li> </ul>	
	5. Session Layer	<ul style="list-style-type: none"> <li>- 송수신지의 프로세스 간 연결 관리</li> <li>- 통신 오류 복구 및 재전송</li> </ul>	
	4. Transport Layer	<ul style="list-style-type: none"> <li>- 세그먼트 구성</li> <li>- 송수신지의 포트(Port) 지정</li> <li>- 메시지 분할 및 재조립</li> <li>- 프로세스 간 혼잡제어, 흐름제어</li> <li>- 오류제어 및 재전송</li> </ul>	로드밸런서, 방화벽
Data Flow Layer (하위 계층)	3. Network Layer	<ul style="list-style-type: none"> <li>- 패킷 구성</li> <li>- 송수신지의 논리 주소(IP) 지정</li> <li>- 최적 경로 탐색 및 전송</li> </ul>	라우터, L3 스위치
	2. Data Link Layer	<ul style="list-style-type: none"> <li>- 프레임 구성</li> <li>- 오류제어, 흐름제어, 접근제어</li> <li>- 송수신지의 물리주소(MAC) 지정</li> </ul>	NIC, L2 스위치
	1. Physical Layer	<ul style="list-style-type: none"> <li>- 비트 스트림의 전기 신호 전송</li> <li>- 비트의 부호화 및 복호화</li> <li>- 물리적 연결 설정 및 해제</li> </ul>	하브, 리피터, 케이블

## 2. TCP/IP(Transmission Control Protocol/Internet Protocol) 프로토콜 스택

### (1) TCP/IP 프로토콜 스택의 개념

OSI 7 Layer를 실무에 활용하는 기능 중심으로 4계층으로 구조화하고, 각 그룹에서 활용되는 프로토콜군을 정리한 네트워크 통신 구조 모델이다.

### (2) TCP/IP 프로토콜 스택의 구성

OSI 7 Layer의 1, 2계층과 5, 6, 7계층을 통합하여 4계층으로 구성된다.

〈표 1-2〉 TCP/IP 프로토콜 스택 구성

OSI 7 Layer	TCP/IP	프로토콜
7. Application		- HTTP(Hypertext Transfer Protocol): 인터넷 브라우저용 - FTP(File Transfer Protocol): 파일 송수신용
6. Presentation	Application	- IMAP(Internet Messaging Access Protocol): 메일 송수신용 - SMTP(Simple Mail Transfer Protocol): 메일 송수신용
5. Session		- TELNET(Telecommunication Network): 터미널 접속용
4. Transport	Transport	- TCP(Transmission Control Protocol): 패킷 송수신용 - UDP(User Datagram Protocol): 패킷 단방향 송신용
3. Network	Internet	- ARP(Address Resolution Protocol): IP의 MAC 주소 변환용 - ICMP(Internet Control Message Protocol): NW 제어용
2. Data Link	Network Access	- RS-232(Recommended Standard-232): 직렬 포트용 - V.35: 케이블 랜선용
1. Physical		- FDDI(Fiber Distributed Data Interface): 광섬유 케이블용

## ② 인캡슐레이션(Encapsulation)과 디캡슐레이션(Decapsulation)

네트워크는 다수의 기기가 연결되어 통신하기 때문에 하나의 기기가 네트워크를 단독으로 점유하지 않고 다수의 기기가 공유하여 동시에 사용할 수 있어야 한다. 이를 위해 송신지에서는 인캡슐레이션 과정을 통해 전송 데이터를 패킷으로 분할하고, 각 패킷에 송수신지의 IP/Port, 전송 순번 등을 같이 보내고 수신지에서 디캡슐레이션 과정을 통해 분할되어 도착된 패킷들을 재조합하여 활용한다.

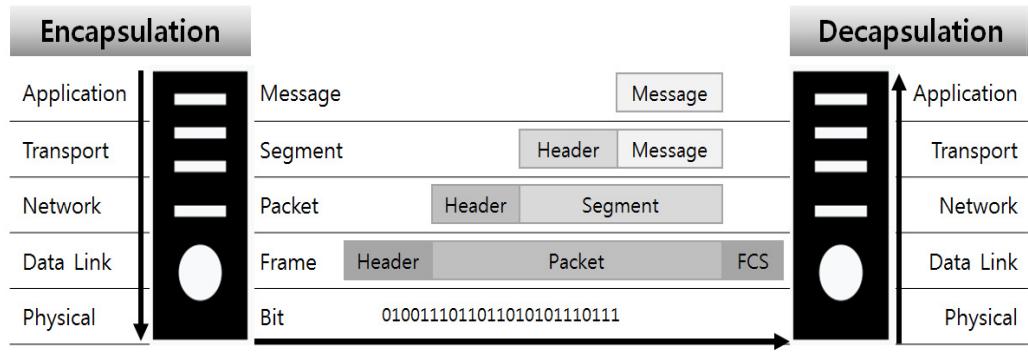
### 1. 인캡슐레이션과 디캡슐레이션의 개념

#### (1) 인캡슐레이션의 개념

송신지 Application Layer에서 발생한 데이터를 하위 계층으로 이동시키면서 각 계층에서 처리한 결과를 캡슐화하는 과정

#### (2) 디캡슐레이션의 개념

수신지의 하위 계층에서 인식한 데이터를 상위 계층으로 이동시키면서 각 네트워크 계층에서 처리 가능한 형태로 디캡슐화하는 과정



출처: 집필진 제작(2021)  
 [그림 1-2] 인캡슐레이션, 디캡슐레이션 개념도

## 2. 인캡슐레이션과 디캡슐레이션의 처리 방법

### (1) 송신지에서의 인캡슐레이션 처리

- Transport 계층에서는 상위 Application 계층에서 생성된 메시지를 전송이 용이하도록 분할하여 Body 정보에 추가하고 수신지에서 분할되어 수신된 메시지를 재조립할 수 있도록 전송 순번, 출발지와 도착지의 Port 정보 등을 Header 정보에 추가하여 Segment를 생성한다.
- Network 계층에서는 상위 Transport 계층에서 생성한 Segment를 Body 정보에 추가하고 출발지와 도착지의 IP 정보 등을 Header에 추가하여 Packet을 생성한다.
- Data Link 계층에서는 상위 Network 계층에서 생성한 Packet을 Body 정보에 추가하고 출발지와 도착지의 MAC 정보 등을 Header에 추가하고 수신지에서 수신된 데이터가 정상인지 판단할 수 있도록 FCS(Frame Check Sequence)를 마지막에 추가하여 Frame을 생성한다.
- Physical 계층에서는 상위 Data Link 계층에서 생성한 Frame을 물리적인 전기 신호로 부호화하여 수신지에 전송한다.

### (2) 수신지에서의 디캡슐레이션 처리

- Physical 계층에서 전기 신호를 비트로 복호화하고 Network 계층에 전달한다.
- Data Link 계층에서 헤더 정보를 체크하여 목적지의 MAC 주소가 자신의 MAC 주소와 동일하면 이더 타입(Ether Type)에 정의되어 있는 Network 계층의 프로토콜로 헤더를 제거한 Packet을 전달한다.
- Network 계층에서 헤더 정보를 체크하여 목적지의 IP 주소가 자신의 IP 주소와 동일하면 프로토콜 번호(Protocol Number)에 정의되어 있는 Transport 계층의 프로토콜로 헤더를 제외한 Segment를 전달한다.
- Transport 계층에서 헤더 정보를 체크하여 도착한 Segment들을 재조립하여

Message를 생성하고 포트 번호(Port Number)에 정의된 Application 계층의 프로토콜로 헤더를 제외한 Message를 전달한다.

#### PDU(Protocol Data Unit)

PDU는 각 네트워크 계층에서 사용하는 데이터 단위를 의미하는 용어로 통신에 필요한 헤더 정보(제어 정보)와 데이터를 캡슐화하여 구성하며 Frame, Packet, Segment 등 계층 별로 다른 단위를 사용한다. 예를 들어 4계층에서는 Segment 단위를 사용하며 송신지와 수신지의 4계층에서는 Segment를 이용해 통신을 수행하게 된다.

### ③ 네트워크 계층별 헤더 정보

송신지와 수신지의 각 계층에서는 독립적으로 통신을 수행하기 위해 PDU의 헤더 정보에 계층별 주요한 제어 정보를 포함한다.

〈표 1-3〉 네트워크 계층별 헤더 정보

OSI 7 Layer	헤더 정보
4. Transport	<ul style="list-style-type: none"> <li>- Sort Port, Destination Port</li> <li>- Sequence Number, Acknowledgement Number</li> <li>- Data Offset, Res, Flags, Window Size, Header and Data Checksum</li> <li>- Urgent Pointer, Options</li> </ul>
3. Network	<ul style="list-style-type: none"> <li>- Source IP, Destination IP, Protocol</li> <li>- Version, IHL, DSCP, ECN, Total Length, Identification</li> <li>- Flags, Fragment Offset, Time to Live, Header Checksum, Options</li> </ul>
2. Data Link	<ul style="list-style-type: none"> <li>- Destination MAC, Source MAC, Ether Type</li> </ul>

〈표 1-4〉 네트워크 상위 프로토콜 지시자

OSI 7 Layer	지시자	주요 상위 프로토콜 값
4. Transport	Port Number	<ul style="list-style-type: none"> <li>- TCP 20, 21: FTP(File Transfer Protocol)</li> <li>- TCP 22: SSH(Secure Shell)</li> <li>- TCP 23: TELNET(Telnet Terminal)</li> <li>- TCP 80, UDP 80: HTTP(Hypertext Transfer Protocol)</li> <li>- TCP 443: HTTPS(Hypertext Transfer Protocol Secure)</li> </ul>
3. Network	Protocol Number	<ul style="list-style-type: none"> <li>- 1: ICMP(Internet Control Message)</li> <li>- 2: IGMP(Internet Group Management)</li> <li>- 6: TCP(Transmission Control)</li> <li>- 17: UDP(User Datagram)</li> </ul>
2. Data Link	Ether Type	<ul style="list-style-type: none"> <li>- Ox0800: IPv4(Internet Protocol Version 4)</li> <li>- Ox86DD: IPv6(Internet Protocol Version 6)</li> <li>- Ox8035: RARP(Reverse ARP)</li> </ul>

#### ④ 라우팅 유형

데이터가 송신지에서 수신지까지 이동하는 경로를 형성하는 것을 라우팅이라고 하며, 크게 정적 라우팅과 동적 라우팅으로 분류한다.

##### 1. 정적 라우팅(Static Routing)

네트워크 관리자가 직접 라우팅 테이블에 라우팅 경로를 입력하여 관리하는 방법을 정적 라우팅이라고 한다.

##### 2. 동적 라우팅(Dynamic Routing)

네트워크 관리자가 직접 개입하지 않고 라우터 간에 정보를 교환하면서 라우팅 경로를 관리하는 방법을 동적 라우팅이라고 한다. 동적 라우팅은 거리 벡터 라우팅 알고리즘과 링크 상태 라우팅 알고리즘을 사용한다.

###### (1) 거리 벡터 라우팅 알고리즘(Distance Vector Routing Algorithm)

인접 라우터와 라우팅 테이블 정보를 교환하여 패킷 전송을 위해 거쳐야 하는 라우터의 개수로 거리를 산정하여 최적의 라우팅 경로를 수립하는 방법으로 RIP 등의 프로토콜을 이용해 라우터 간 정보를 교환한다.

###### (가) RIP(Routing Information Protocol)

라우팅 테이블에 동일 네트워크에 포함된 각 라우터에 도달하기 위해 거쳐야 하는 라우터들의 최대 수와 각 라우터에 도달하기 위해 이동해야 하는 다음 라우터 정보를 관리하고, 정기적으로 라우팅 테이블 정보를 인접 라우터와 교환하여 자신의 라우팅 테이블을 갱신하는 방법이다.

###### (2) 링크 상태 라우팅 알고리즘(Link State Routing Algorithm)

연결된 모든 라우터로부터 연결 상태 정보를 수신하여 각 라우터까지 최단 경로를 라우팅 테이블로 만드는 방법으로, OSPF 등의 프로토콜을 이용해 라우터 간 정보를 교환한다.

###### (가) OSPF(Open Shortest Path First)

라우터의 연결 상태가 변경된 경우 동일 네트워크에 포함된 모든 라우터에 자신의 변경 정보를 전달하고, 정보를 수신한 라우터들이 각 라우터에 접근하기 위한 최적의 네트워크 경로를 갱신하는 방법이다.

## 수행 내용 / 네트워크 프로토콜 활용하기

---

### 재료 · 자료

- 네트워크 구성도
- 요구사항 명세서
- 테스트 가능한 응용소프트웨어 URL

### 기기(장비 · 공구)

- 라우터
- 오픈소스 기반 네트워크 패킷 분석 프로그램
- 오픈소스 기반 FTP Client 프로그램

### 안전 · 유의 사항

- 학습용 소프트웨어는 라이선스 정책을 확인하여 저작권에 위배되지 않도록 선택한다.
- 시스템 상태 체크 등의 학습 시 정상 운영 중인 시스템에 부하를 주지 않도록 주의한다.
- 타 시스템 이용한 학습 시 비인가 접근 등 해킹으로 오해받지 않도록 주의한다.

### 수행 순서

#### ① 네트워크 계층 구조를 학습하고 각 계층의 역할을 구별한다.

응용소프트웨어 개발을 위해서는 적절한 프로토콜과 네트워크 시스템 및 장비를 활용할 수 있도록 네트워크 계층 구조를 이해하고 각 계층별 역할을 구별할 수 있어야 한다. 이번 과정에는 각 네트워크 계층 구조를 학습하고 각 계층의 역할을 구별한다.

##### 1. 네트워크 계층 구조 및 계층별 역할을 학습한다.

###### (1) OSI 7 Layer 참조 모델을 학습한다.

데이터의 통신 과정을 이해하기 위해 학습 내용 1-1의 필요 지식 ①-1에서 설명 한 OSI 7 Layer 참조 모델을 학습한다.

###### (2) TCP/IP 프로토콜 스택을 학습한다.

실무에 활용되는 네트워크 계층별 구조를 파악하기 위해 학습 내용 1-1의 필요 지식 ①-2에서 설명한 TCP/IP 프로토콜 스택을 학습한다.

###### (3) 인캡슐레이션과 디캡슐레이션을 학습한다.

네트워크 계층별로 독립적으로 통신을 하기 위해 송수신지에서 수행하는 인캡슐레이션과 디캡슐레이션 과정을 학습 내용 1-1의 필요 지식 ②를 참고하여 학습한다.

(4) 네트워크 계층별 역할을 수행하기 위한 제어 정보가 포함된 헤더 구조를 학습한다.

송신지에서 네트워크 계층별로 인캡슐레이션하여 Physical 계층을 통해 송신한 데이터는 수신지의 동일한 네트워크 계층에서 디캡슐레이션되면서 송수신지의 동일한 네트워크 계층 간에 통신이 이루어진다. 각 계층을 제어하기 위한 설정 정보는 각 계층의 헤더 정보에 포함되므로 학습 내용 1-1의 필요 지식 ③을 참고하여 네트워크 계층별 헤더 구조와 프로토콜을 학습한다.

2. 네트워크 계층의 헤더 정보를 분석하여 네트워크 계층의 역할을 구별한다.

네트워크 계층별로 생성되는 헤더 정보를 직접 분석해 보면서 네트워크 계층별로 수행되는 역할을 구별한다.

(1) 네트워크 패킷 분석기를 설치한다.

네트워크로 송수신되는 헤더 정보의 구조를 분석하기 위해 오픈소스 네트워크 패킷 분석 프로그램인 Wireshark를 설치한다.

**수행 tip**

- 네트워크 프로토콜 분석기는 Wireshark, Tcpdump 등 다양한 프로그램이 존재하나 이번 학습에서는 윈도 환경에 적합한 GUI(Graphic User Interface) 기반의 Wireshark를 이용한다.
- 다운로드: <https://www.wireshark.org/#download>
- 컴퓨터 환경에 따라 설치 파일이 다르므로 본인의 컴퓨터 환경에 맞춰서 다운로드 후 설치하고 사용자 가이드도 다운로드 받아서 학습한다.

(2) 국가직무능력표준 시스템의 IP를 확인한다.

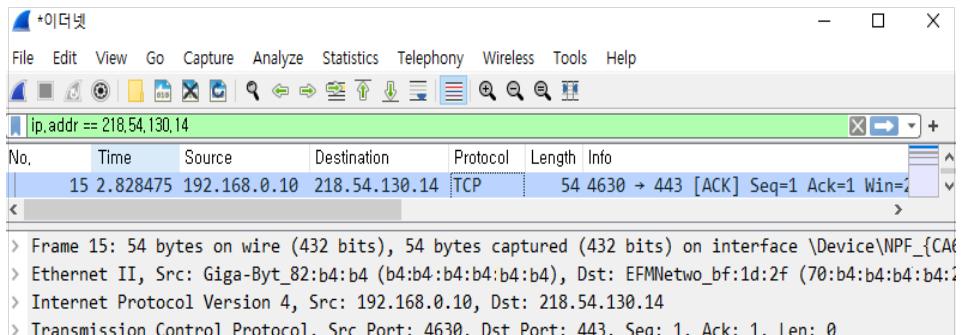
도메인 네임으로 IP 주소를 확인할 수 있는 nslookup(name server lookup) 명령어를 이용해 ncs.go.kr의 아이피를 확인한다.

**Command**

```
C:\Windows\System32>nslookup ncs.go.kr  
서버:      kms.kornet.net  
Address:   168.126.63.1
```

```
권한 없는 응답:  
이름:      ncs.go.kr  
Address:   218.54.130.14
```

- (3) 네트워크 패킷 분석기로 계층별 헤더 정보를 분석하여 계층별 역할을 확인한다.
- (가) 네트워크 패킷 분석 프로그램을 실행한다.
  - (나) 웹 브라우저에서 국가직무능력표준 시스템에 접속한다.
  - (다) 화면 상단에서 Stop Capturing Packets 아이콘을 클릭하여 캡처를 중단한다.
  - (라) 화면 상단 입력창에 ip.addr == 218.54.130.14를 입력하여 국가직무능력표준 시스템의 IP만 필터링하여 조회되도록 설정한다.
  - (마) 필터링된 결과 중 TCP 프로토콜을 클릭하여 하단에 표시되는 계층별 헤더 정보를 분석하여 네트워크 계층별 역할을 수행하기 위한 제어 정보들을 확인한다.
    - 1) [그림 1-4]와 같이 3계층 헤더 정보에서 송신지 IP를 Source Address에서 확인하고 수신지 IP를 Destination Address에서 확인할 수 있으며, 상위 4 계층의 Protocol이 TCP라는 정보 등을 확인한다.
    - 2) [그림 1-5]와 같이 4계층 헤더 정보에서 Source Port, Destination Port 등을 확인하고, Port가 4430으로 SSL이 적용된 HTTPS 프로토콜을 사용한다는 것을 확인한다.



출처: 집필진 제작(2021)  
[그림 1-3] 네트워크 패킷 분석기 예시

```

▼ Internet Protocol Version 4, Src: 192.168.0.10, Dst: 218.54.130.14
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 40
    Identification: 0xcd0e (52494)
    > Flags: 0x40, Don't fragment
      Fragment Offset: 0
      Time to Live: 128
      Protocol: TCP (6)
      Header Checksum: 0x0000 [validation disabled]
      [Header checksum status: Unverified]
      Source Address: 192.168.0.10
      Destination Address: 218.54.130.14
  
```

출처: 집필진 제작(2021)  
[그림 1-4] 3계층 헤더 정보 예시

```

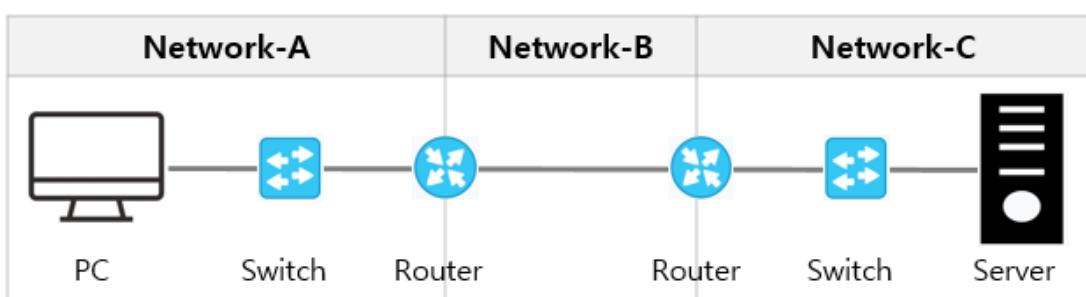
▼ Transmission Control Protocol, Src Port: 4630, Dst Port: 443, Seq: 1, Ack: 1, Len: 0
  Source Port: 4630
  Destination Port: 443
  [Stream index: 2]
  [TCP Segment Len: 0]
  Sequence Number: 1      (relative sequence number)
  Sequence Number (raw): 2750486480
  [Next Sequence Number: 1      (relative sequence number)]
  Acknowledgment Number: 1      (relative ack number)
  Acknowledgment number (raw): 634702437
  0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
  Window: 1024
  [Calculated window size: 262144]
  [Window size scaling factor: 256]
  Checksum: 0x1d12 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  > [SEQ/ACK analysis]
  > [Timestamps]

```

출처: 집필진 제작(2021)  
[그림 1-5] 4계층 헤더 정보 예시

## ② 패킷 스위칭 시스템의 구조를 파악하고 라우팅 알고리즘과 프로토콜을 식별한다.

일반적으로 사용자 컴퓨터에서 원거리의 서버에 접근하기 위해서는 [그림 1-6]과 같이 다수의 네트워크를 통과하게 되며, 각 네트워크는 중계 장치로 연결된다. 이런 중계 장치를 라우터(Router)라고 하며, 앞서 학습했던 네트워크 3계층에서 라우터가 송수신지의 IP를 통해 제어한다. 응용소프트웨어 개발환경에서 네트워크 문제가 발생하는 경우 원인을 분석하고 해결할 수 있도록 패킷 스위칭 시스템의 구조를 파악하고 라우팅 알고리즘과 프로토콜을 식별한다.



출처: 집필진 제작(2021)  
[그림 1-6] Router를 통한 네트워크 연결 구조 예시

### 패킷 스위칭 시스템(Packet Switching System)

데이터를 패킷으로 분할하고 송수신 주소를 헤더로 만들어 교환기가 헤더 정보를 참고하여 수신자까지 전달시키는 데이터 통신 방식의 시스템이다.

## 1. 패킷 스위칭 시스템의 구조를 파악한다.

학습자마다 네트워크 환경이 다르므로 학습자의 IP 정보를 확인하고 서버에 연결되는 과정을 확인하여 패킷 스위칭 시스템 구조를 파악한다.

### (1) 학습자의 IP와 기본 라우터 정보를 확인한다.

(가) 명령어 처리기 프로그램을 실행한다.

(나) ipconfig /all 명령어를 입력한다.

ipconfig 명령어로 NIC(Network Interface Card) 설정 정보를 확인한다.

(다)Ipv4 주소로 할당된 IP 정보를 확인한다.

(라) 기본 게이트웨이로 기본 라우터 IP를 확인한다.

목적지 IP가 라우터에 등록되지 않은 경우 기본으로 전송하는 라우터의 IP가 지정되어 있는 기본 게이트웨이를 확인한다.

### Command

```
C:\Windows\System32>ipconfig /all  
Windows IP 구성  
호스트 이름 . . . . . : test  
주 DNS 접미사 . . . . . :  
노드 유형 . . . . . : 혼성  
IP 라우팅 사용. . . . . : 아니요  
WINS 프록시 사용. . . . . : 아니요  
이더넷 어댑터 이더넷:  
연결별 DNS 접미사. . . . . :  
설명. . . . . . . . . . . : Intel(R) Ethernet Connection (7) I219-V  
물리적 주소 . . . . . : B4-B4-B4-B4-B4-B4  
DHCP 사용 . . . . . : 예  
자동 구성 사용. . . . . : 예  
링크-로컬 IPv6 주소 . . . . . : fe80::fe80:fe80:d85b%16(기본 설정)  
IPv4 주소 . . . . . : 192.168.0.10(기본 설정)  
서브넷 마스크 . . . . . : 255.255.255.0  
임대 시작 날짜. . . . . : 2021년 6월 12일 토요일 오후 12:28:25  
임대 만료 날짜. . . . . : 2021년 6월 12일 토요일 오후 5:28:25  
기본 게이트웨이 . . . . . : 192.168.0.1  
DHCP 서버 . . . . . : 192.168.0.1  
DHCPv6 IAID . . . . . : 62140057  
DHCPv6 클라이언트 DUID. . . . . :  
00-01-00-01-25-32-40-FD-B4-2E-99-82-E6-1F  
DNS 서버. . . . . : 168.126.63.1  
168.126.63.2  
Tcpip를 통한 NetBIOS. . . . . : 사용
```

(2) 접속하고자 하는 시스템의 네트워크 연결 상태를 확인한다.

ping(Packet InterNet Groper) 명령어로 접속하고자 하는 시스템의 IP 또는 URL을 입력하여 접속 여부를 확인한다. 네트워크 연결이 정상이면 왕복 시간이 표시되고, 네트워크 연결이 비정상이면 호스트를 찾을 수 없다고 표시되거나 요청 시간이 만료되었다고 표시된다. 이 경우 입력 정보가 잘못되었는지, 네트워크 연결 상태에 문제가 없는지 확인하고 문제가 없는 경우 라우팅 경로에 있는 모든 장비의 연결 상태를 순차적으로 점검한다.

**Command**

```
C:\Windows\System32>ping ncs.go.kr
Ping ncs.go.kr [218.54.130.14] 32바이트 데이터 사용:
218.54.130.14의 응답: 바이트=32 시간=10ms TTL=241
218.54.130.14의 응답: 바이트=32 시간=10ms TTL=241
218.54.130.14의 응답: 바이트=32 시간=10ms TTL=241
218.54.130.14의 응답: 바이트=32 시간=9ms TTL=241
218.54.130.14에 대한 Ping 통계:
    패킷: 보냄 = 4, 받음 = 4, 손실 = 0 (0% 손실),
왕복 시간(밀리초):
    최소 = 9ms, 최대 = 10ms, 평균 = 9ms
```

```
C:\Windows\System32>ping nn.go.kr
Ping 요청에서 nn.go.kr 호스트를 찾을 수 없습니다. 이름을 확인하고 다시 시도하십시오.
```

```
C:\Windows\System32>ping 218.55.130.14
Ping 218.55.130.14 32바이트 데이터 사용:
요청 시간이 만료되었습니다.
요청 시간이 만료되었습니다.
요청 시간이 만료되었습니다.
218.55.130.14에 대한 Ping 통계:
    패킷: 보냄 = 4, 받음 = 0, 손실 = 4 (100% 손실),
```

(3) 목적지 시스템까지의 라우팅 경로를 확인하여 네트워크 상태를 파악한다.

tracert 명령어로 접속하고자 하는 시스템까지 경유하는 라우터들을 확인한다. 라우팅 경로에 문제가 있거나 중간 라우터에 장애가 있는 경우 또는 방화벽 정책에 의해 연결이 막혀 있는 경우 목적지 IP까지 도달하지 못한다.

### Command

```
C:\Windows\System32>tracert 218.54.130.14
최대 30홉 이상의
ncs.go.kr [218.54.130.14](으)로 가는 경로 추적:
  1  <1 ms    <1 ms    <1 ms  192.168.0.1
  2      2 ms      *      2 ms  210.113.56.1
~ 중간 생략 ~
  15     10 ms     9 ms    9 ms  218.54.130.14
추적을 완료했습니다.
```

### 수행 tip

- 윈도는 ipconfig 명령어를 사용하고 유닉스/리눅스 계열은 ifconfig 명령어를 사용한다.
- 윈도는 tracert 명령어를 사용하고 유닉스/리눅스 계열은 traceroute 명령어를 사용한다.
- ipconfig, tracert 등 대부분의 명령어는 ? 옵션으로 각 명령어의 옵션 종류를 확인할 수 있다.  
(예시: ipconfig /? )
- 기본 게이트웨이가 0.0.0.0 이면 외부 네트워크와 단절된 상태이다.
- 네트워크의 그루핑(Grouping) 현황을 알기 위해서는 서브넷 마스크도 중요하므로 별도로 학습하도록 한다.

## 2. 라우팅 알고리즘과 프로토콜을 식별한다.

라우팅 경로 설정의 상세 현황을 파악하여 라우팅 알고리즘과 프로토콜을 식별하기 위해 라우터에 접속하여 라우팅 테이블을 조회한다. 목적지 시스템까지 연결이 원활하지 않은 경우 라우터에 설정되어 있는 라우팅 테이블의 정보를 확인하여 문제 지점을 찾고 네트워크 엔지니어와 협업하여 문제를 해결할 수 있다.

### (1) 라우팅 장비에 접속한다.

라우팅 테이블 정보를 확인하기 위해 라우터 또는 라우팅을 수행할 수 있는 멀티레이어 스위치에 접근한다.

#### 라우팅 테이블(Routing Table)

라우터에 전달된 패킷이 목적지까지 전달되기 위해 거쳐야 하는 라우팅 경로를 관리하는 테이블로 관리자가 직접 입력하여 관리하거나 라우터 간에 정보를 공유하면서 상황에 따라 동적으로 재구성한다. 전자의 경우를 Static Routing이라고 하며 네트워크 장애가 있는 경우에도 관리자가 직접 대체 경로를 입력해줘야 하므로 변화가 적은 소규모 네트워크에 적합하다. 후자의 경우는 Dynamic Routing이라고 하며 대부분의 일반적인 네트워크에서 사용한다.

## (2) 라우팅 테이블을 조회하여 라우팅 알고리즘과 프로토콜을 식별한다.

라우터 장비의 제조사에 따라 라우팅 테이블을 조회하는 명령어가 다르므로 명령어를 확인하여 라우팅 테이블을 조회한다. 아래 예시는 C사의 라우터에서 조회한 라우팅 테이블로 라우팅 프로토콜, 라우트 타입, 네트워크 ID, 라우팅 metric, 다음 라우터 주소, 마지막 갱신 시간 등을 확인할 수 있다. 라우팅 알고리즘과 프로토콜 종류에 대해서는 학습 내용 1-1의 필요 지식 ④ 라우팅 유형을 참고한다.

### Command

```
route-svr> show ip route
Codes: I - IGRP derived, R - RIP derived, O - OSPF derived
      C - connected, S - static, E - EGP derived, B - BGP derived
      * - candidate default route, IA - OSPF inter area route
      E1 - OSPF external type 1 route, E2 - OSPF external type 2 route
Gateway of last resort is 11.11.1.1 to network 120.100.0.0
O E2 150.150.0.0 [260/5] via 131.120.254.16, 0:01:00, Ethernet2
E     192.67.131.0 [300/228] via 131.120.254.244, 0:02:22, Ethernet2
O E2 192.68.132.0 [260/5] via 131.120.254.16, 0:00:59, Ethernet2
O E2 130.130.0.0 [360/5] via 131.120.254.16, 0:00:59, Ethernet2
E     128.128.0.0 [300/228] via 131.120.254.244, 0:02:22, Ethernet2
~ 이하 생략 ~
```

## ③ 응용소프트웨어 특성을 파악하고 프로토콜을 구별하여 적용한다.

응용소프트웨어 개발을 위해서는 전송 계층, 애플리케이션 계층의 다양한 프로토콜을 사용하게 된다. 응용소프트웨어의 특성을 파악하고 필요한 프로토콜을 구별하여 적용해 본다.

### 1. 응용소프트웨어의 특성을 파악한다.

응용소프트웨어는 사용자 및 기획자의 목적에 맞도록 만든 시스템으로 구현되어야 하는 기능들을 요구사항 명세서로 작성하므로 요구사항 명세서를 분석하여 응용소프트웨어의 특성을 파악한다.

#### (1) 아래 <표 1-5>의 요구사항 명세서 예시를 확인한다.

요구사항 명세서는 응용소프트웨어 구축 기획 단계에서 기획자가 초안을 작성하며, 구축 프로젝트가 시작되면 분석/설계 담당자가 기획자와 구현 방안 등을 협의하여 확정하고 이를 토대로 프로그램을 설계한다. 본 예시는 Web 환경으로 구축하는 주문관리 시스템의 요구사항 명세서 일부를 발췌한 자료이다.

#### (2) 요구사항을 분석하고 구현 기능의 특성을 파악한다.

예시를 통해 네트워크 구간 암호화에 대한 보안 요구사항과 주문관리, 납품관리 등 기능 요구사항이 포함되어 있음을 확인한다. 각 기능 구현을 위해 필요한 사항을 상세하게 도출해본다.

〈표 1-5〉 요구사항 명세서 예시

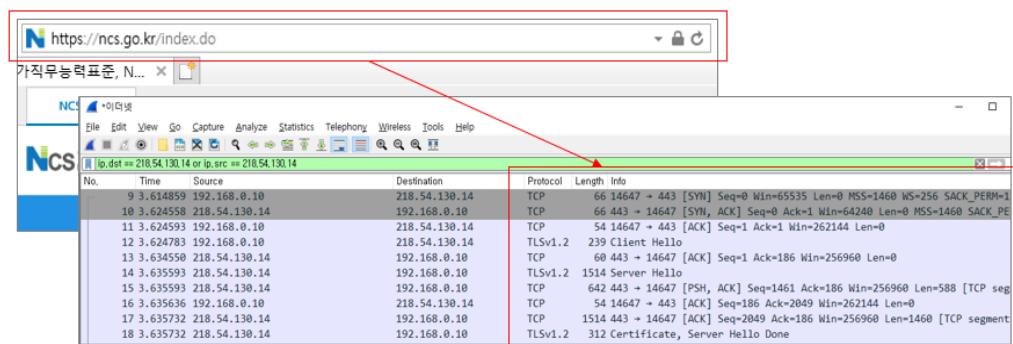
RQ ID	화면	요구사항	상세내역	요청일	요청자	상태
RQ-01	공통	보안	네트워크 구간 암호화	6/13	홍길동	확정
RQ-02	주문관리	주문서 작성	거래처, 품목, 수량, 금액 저장	6/13	홍길동	확정
RQ-03	주문관리	주문서 취소	입력된 주문서를 조회하여 삭제	6/13	홍길동	확정
RQ-04	주문관리	주문서 발송	주문서 메일 발송	6/13	홍길동	확정
RQ-05	납품관리	납품서 작성	납품일, 수량, 납품 바코드 저장	6/15	홍길동	확정
RQ-06	납품관리	납품서 업로드	납품서 파일서버 전송 저장	6/15	홍길동	협의중

## 2. 기능 구현을 위해 필요한 프로토콜을 파악한다.

〈표 1-5〉 요구사항 명세서 예시의 기능 구현을 위해 필요한 프로토콜을 파악한다.

### (1) RQ-01의 네트워크 구간 암호화 적용을 위해 HTTPS 프로토콜을 파악한다.

네트워크 구간 암호화는 사용자의 웹 브라우저와 웹 서버 사이에 송수신 중인 데이터를 암호화하여 중간에 데이터를 탈취하여 악용하는 것을 방지하기 위한 기법으로 HTTPS(HyperText Transfer Protocol over Secure socket layer) 프로토콜을 이용하여 기능을 구현할 수 있다. HTTPS 프로토콜을 이용하기 위해서는 웹 서버에 SSL 인증서를 설치하고 사용자가 HTTP로 접속을 하면 HTTPS로 Redirect하도록 설정해야 한다. HTTPS는 TCP 443 포트를 이용해 통신하는 프로토콜로 아래 [그림 1-7]과 같이 패킷 분석 도구를 이용해 확인할 수 있다.



출처: 집필진 제작(2021)

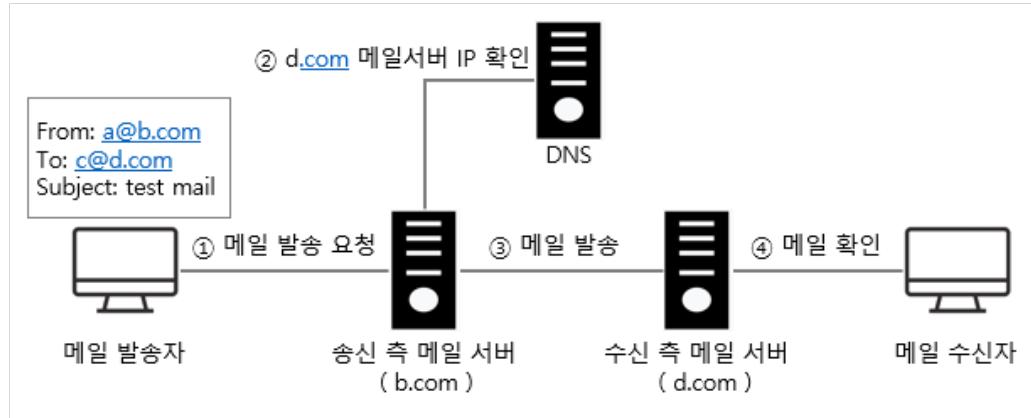
[그림 1-7] HTTPS 프로토콜 패킷 분석 예시

### 수행 tip

- TLS는 SSL3.0의 업그레이드 버전으로 SSL의 보안 취약점을 보완한 네트워크 보안 프로토콜이다.

### (2) RQ-04의 메일 발송 기능을 위해 SMTP를 파악한다.

메일 발송을 위해서는 TCP 25번 포트를 사용하는 애플리케이션 계층 프로토콜인 SMTP(Simple Mail Transfer Protocol)를 사용하며, [그림 1-8]과 같은 구조로 메일을 발송한다. 이 기능을 사용하기 위해서는 웹 서버에 SMTP 전자 메일 서비스 설정을 해야 한다.



출처: 집필진 제작(2021)  
[그림 1-8] 메일 발송 구조

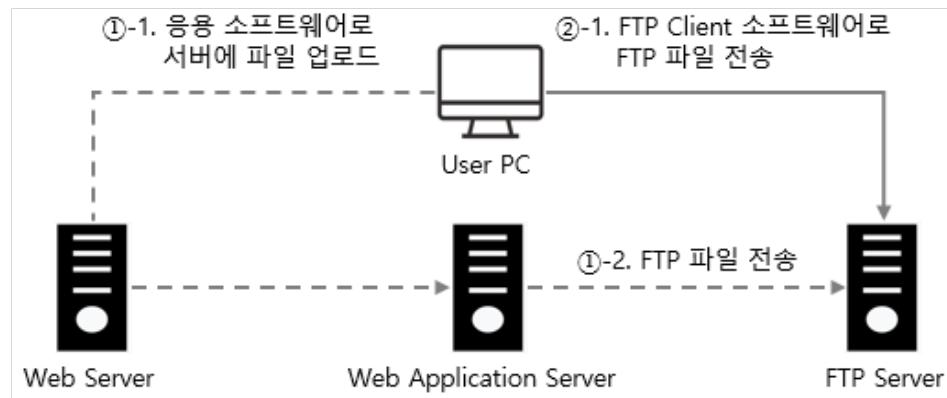
### 수행 tip

- 서버 환경 구성 시 방화벽에서 25번 포트의 사용을 차단했을 수 있으므로 방화벽 해제를 네트워크 담당자에 요청해야 한다.
- 25번 포트가 일반적이지만 최근에는 보안을 위해 587 번 포트를 사용하는 추세이다.
- 메일 서버에서 사용자 로컬 환경으로 메일을 수신하기 위해서는 POP3 프로토콜을 사용한다.

### (3) RQ-06의 파일 서버 전송 기능을 위해 FTP를 확인한다.

파일 서버로 자료를 관리하려면 [그림 1-9]와 같이 응용소프트웨어로 WAS(Web Application Server)에 파일을 업로드하고 WAS에서 파일 서버로 FTP(File Transfer Protocol) 전송하도록 프로그램을 개발하거나 FTP Client 소프트웨어로 FTP Server에 직접 접속하여 업로드한다. 이때 사용하는 FTP는 애플리케이션 계층의 파일 전송용 프로토콜로 능동 모드(Active Mode)와 수동 모드(Passive Mode)가 있으며 각각 사용하는 포트가 다르다. 능동 모드는 FTP Server 접속을 위한 21

번 포트와 데이터 전송을 위한 20번 포트를 사용하고 수동 모드는 21번 포트는 동일하게 사용하지만, 데이터 전송용 포트는 1024 이상의 포트를 FTP Server가 정해서 사용한다. 파일 서버에 FTP 사용을 위해서 FTP 설정해야 한다.



출처: 집필진 제작(2021)  
[그림 1-9] FTP 파일 전송 방법

#### 수행 tip

- FTP는 보안에 취약하여 최근에는 암호화하여 파일을 송수신하는 sFTP를 사용하는 추세이다. sFTP는 22번 포트를 사용한다.
- FTP 유형 및 모드에 따라 방화벽 해제가 필요하다.

### 3. 프로토콜을 적용한다.

파악된 프로토콜을 적용하기 위해서는 서버 설정과 네트워크 방화벽 해제 작업을 해야 한다. 예제로 파악한 프로토콜 중 FTP를 이용하기 위한 서버 설정을 진행하고 네트워크 상태 체크 및 방화벽 해제 실습을 한다.

#### (1) 리눅스 서버에 접속하여 FTP 소프트웨어를 설치한다.

리눅스 환경에서 오픈소스 소프트웨어인 vsftpd를 이용해 실습한다.

##### (가) APT 파일 목록을 갱신한다.

###### Command

```
ubuntu-server@ubuntu-server:~$ sudo apt update
```

##### (나) vsftpd 설치 패키지 존재 여부를 체크한다.

###### Command

```
ubuntu-server@ubuntu-server:~$ sudo apt search vsftpd
```

##### (다) vsftpd를 설치한다.

설치하면 자동으로 서비스가 등록되고 실행된다.

#### Command

```
ubuntu-server@ubuntu-server:~$ sudo apt install vsftpd
```

(2) FTP 소프트웨어 설치 상태를 점검한다.

(가) vsftpd의 실행 상태를 확인한다.

#### Command

```
ubuntu-server@ubuntu-server:~$ sudo service vsftpd status
● vsftpd.service - vsftpd FTP server
   Loaded: loaded (/lib/systemd/system/vsftpd.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2021-06-19 11:38:12 UTC; 9min ago
     Main PID: 3465 (vsftpd)
        Tasks: 1 (limit: 1078)
       Memory: 620.0K
          CGroup: /system.slice/vsftpd.service
                  └─3465 /usr/sbin/vsftpd /etc/vsftpd.conf

Jun 19 11:38:12 ubuntu-server systemd[1]: Starting vsftpd FTP server...
Jun 19 11:38:12 ubuntu-server systemd[1]: Started vsftpd FTP server.
ubuntu-server@ubuntu-server:~$
```

출처: 집필진 제작(2021)

[그림 1-10] vsftpd 실행 상태 확인 예시

(나) FTP 네트워크 상태를 확인한다.

TCP 21번 포트에서 FTP 접속을 대기 중이다.

#### Command

```
ubuntu-server@ubuntu-server:~$ sudo netstat -natp | grep ftp
tcp6      0      0 ::21          ::*          LISTEN 3465/vsftpd
```

(다) FTP 설정을 확인하고 필요한 부분이 있으면 변경한다.

포트를 변경하거나 FTP Mode에 대한 설정을 변경할 수 있다.

1) /etc/vsftpd.conf 파일을 백업한다.

#### Command

```
ubuntu-server@ubuntu-server:~$
cp /etc/vsftpd.conf /etc/vsftpd.conf.0620bk
```

2) vi 편집기로 /etc/vsftpd.conf 파일을 연다.

#### Command

```
ubuntu-server@ubuntu-server:~$ vi /etc/vsftpd.conf
```

3) #write\_enable=YES의 #을 제거하여 사용자의 파일 쓰기 허용을 설정한다.

4) :wq!를 입력하여 저장하고 vi 편집기를 종료한다.

5) vsftpd 서비스를 재기동한다.

**Command**

```
ubuntu-server@ubuntu-server:~$ sudo systemctl restart vsftpd
```

(3) 20, 21번 포트 방화벽을 해제하고 상태를 확인한다.

**Command**

```
ubuntu-server@ubuntu-server:/usr/bin$ sudo ufw allow 20/tcp
Rules updated
Rules updated (v6)
ubuntu-server@ubuntu-server:/usr/bin$ sudo ufw allow 21/tcp
Rules updated
Rules updated (v6)
ubuntu-server@ubuntu-server:/usr/bin$ sudo ufw enable
Firewall is active and enabled on system startup
ubuntu-server@ubuntu-server:/usr/bin$ sudo ufw status
Status: active

To                         Action      From
--                         --          --
20/tcp                      ALLOW       Anywhere
21/tcp                      ALLOW       Anywhere
20/tcp (v6)                 ALLOW       Anywhere (v6)
21/tcp (v6)                 ALLOW       Anywhere (v6)
```

출처: 집필진 제작(2021)  
[그림 1-11] 방화벽 해제 및 네트워크 상태 확인 예시

(4) FTP 접속용 계정을 생성한다.

**Command**

```
ubuntu-server@ubuntu-server:/usr/bin$ sudo adduser ftptestuser
Adding user `ftptestuser' ...
Adding new group `ftptestuser' (1000) ...
Adding new user `ftptestuser' (1000) with group `ftptestuser' ...
Creating home directory `/home/ftptestuser' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for ftptestuser
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
ubuntu-server@ubuntu-server:/usr/bin$ _
```

출처: 집필진 제작(2021)  
[그림 1-12] FTP 접속 계정 생성 예시

(5) 오픈소스 소프트웨어인 FileZilla를 다운로드 및 설치하여 접속 테스트를 한다.

## 학습1 교수·학습 방법

### 교수 방법

- 실습 전에 OSI 7 계층 등 네트워크 기초 지식에 대해 충분히 이해할 수 있도록 상세히 설명한다.
- 학습에 이용되는 오픈소스 도구들의 라이선스 정책을 같이 확인하며 오픈소스 소프트웨어의 활용에 대해 올바르게 이해하도록 지도한다.
- 학습자들이 동일한 네트워크 환경에서 실습하는 경우 네트워크에 문제가 발생하여 학습 시간에 영향을 줄 수 있으므로 개별적인 가상화 환경을 제공하여 실습하도록 지도한다.
- 운영체제 환경, 네트워크 장비의 벤더 종류 등에 따라 사용하는 명령어 종류 및 사용법이 다르므로 다양한 환경에 따른 차이를 구체적으로 설명하고 스스로 매뉴얼 등을 이용해 학습할 수 있도록 지도한다.

### 학습 방법

- 네트워크 기초 지식을 충분히 습득하도록 이론 학습을 반드시 선행한다.
- 학습 과정에서 사용하는 소프트웨어들의 라이선스 정책을 직접 찾아보고 저작권 등의 이슈가 없는지 점검한다.
- 운영체제 및 네트워크 장비에 따라 사용해야 하는 명령어가 다를 수 있으므로 학습 환경을 먼저 파악하고 학습 자료와 다를 경우 학습 환경에 맞는 자료를 찾아서 보강하여 공유한다.
- 네트워크 패킷 분석기를 통해 다양한 프로토콜의 헤더 정보가 어떻게 구성되는지 파악해 보고 통신 과정에서 변화하는 헤더 정보를 학습하고 궁금한 사항은 질문한다.
- 응용소프트웨어 엔지니어링에서 주로 사용되는 프로토콜을 조사하여 정리하고 그룹 토의를 통해 학습 내용을 공유한다.

## 학습1 평 가

### 평가 준거

- 평가자는 학습자가 학습 목표를 성공적으로 달성하였는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.

학습 내용	학습 목표	성취수준		
		상	중	하
네트워크 프로토콜 활용	- 네트워크 계층 구조에서 각 계층의 역할을 구별할 수 있다.			
	- 패킷 스위칭 시스템의 라우팅 알고리즘과 프로토콜을 식별 할 수 있다.			
	- 응용소프트웨어의 특성에 따라 프로토콜을 구별하여 적용할 수 있다.			

### 평가 방법

- 서술형 시험

학습 내용	평가 항목	성취수준		
		상	중	하
네트워크 프로토콜 활용	- 네트워크 계층 구조에서 각 계층의 역할 구별 능력			
	- 라우팅 알고리즘 및 프로토콜 식별 여부			
	- 응용소프트웨어 특성에 따른 프로토콜 구별 능력			

- 사례 연구

학습 내용	평가 항목	성취수준		
		상	중	하
네트워크 프로토콜 활용	- 응용소프트웨어 엔지니어링에서 주로 활용되는 프로토콜 분석 능력			
	- 네트워크 계층별로 주로 활용되는 프로토콜 숙지 여부			

• 작업장 평가

학습 내용	평가 항목	성취수준		
		상	중	하
네트워크 프로토콜 활용	<ul style="list-style-type: none"> <li>- 네트워크 환경 점검 능력</li> <li>- 네트워크 패킷 분석 도구 활용 능력</li> </ul>			

**피드백**

1. 서술형 시험
  - 네트워크 계층 구조를 잘 이해하고 각 계층의 역할을 명확히 제시했는지 평가하고, 미흡한 부분에 대해서는 주요 내용을 구체적으로 추가 설명한다.
  - 라우팅 알고리즘 및 프로토콜을 식별할 수 있는지 평가하고, 성취수준이 높은 학습자의 경우 다양한 프로토콜에 대해 심화 설명해주고 미흡한 부분에 대해서는 유형별 차이점을 도식화하여 설명해준다.
  - 응용소프트웨어 특성에 따른 프로토콜 구별 능력이 충분한지 평가하고 평가 결과가 미흡한 경우에는 사례를 들어서 추가 설명해준다.
2. 사례 연구
  - 응용소프트웨어 엔지니어링에서 주로 활용되는 프로토콜의 사례를 충분히 제시했는지 평가하고 미흡한 경우에는 주요 사례를 부가적으로 설명해준다.
  - 네트워크 계층별로 주로 활용되는 프로토콜을 숙지하고 있는지 평가하고 미흡한 경우에는 대표 사례를 추가 설명해준다.
3. 작업장 평가
  - 네트워크 환경 점검에 활용 가능한 명령어들과 네트워크 패킷 분석 도구들을 능숙하게 활용할 수 있는지 평가하고, 성취수준이 미흡하면 그룹 활동을 통해 추가 보완한다.

학습 1	네트워크 기초 활용하기
학습 2	미들웨어 기초 활용하기
학습 3	데이터베이스 기초 활용하기

## 2-1. 미들웨어 파악

### 학습 목표

- 운영체제와 응용소프트웨어 사이에 존재하는 미들웨어의 역할을 파악할 수 있다.
- 미들웨어에서 제공하는 기능 및 처리 흐름을 파악할 수 있다.

### 필요 지식 /

#### ① 미들웨어(Middleware)

##### 1. 미들웨어 개념

하나의 시스템에서 다양한 목적의 응용소프트웨어가 동시에 수행되거나 복수 시스템의 응용소프트웨어가 서로 연계되어 수행되는 경우에도 안정적으로 실행될 수 있도록 운영체제와 응용소프트웨어 사이에서 다양한 기능을 지원하는 소프트웨어이다.

##### 2. 미들웨어 주요 기능

미들웨어는 분산 시스템 SW, IT 자원 관리, 서비스 플랫폼 및 네트워크 보안 등의 기능을 포함한다.

〈표 2-1〉 미들웨어 주요 기능

주요기능	설명
분산 시스템 SW	- 물리적으로 분산되어 구축되어 있는 다수의 컴퓨팅 환경에서 사용자가 하나의 시스템처럼 사용할 수 있도록 구성된 소프트웨어
IT 자원 관리	- IT 자원에 대한 관리 정책을 기반으로 지속적으로 모니터링하고 성능과 가용성을 관리하는 기능을 제공하는 소프트웨어
서비스 플랫폼	- 서로 다른 서비스들을 하나의 통합 환경에서 인터랙티브하게 사용할 수 있도록 해주는 인터넷 기반 환경 구성 기술
네트워크 보안	- 네트워크에 연결된 호스트들의 송수신 정보 탈취 및 변조를 통한 불법적인 서비스 이용을 방지하는 기술

### 3. 미들웨어 주요 기능별 분류 체계

#### (1) 분산 시스템 SW

〈표 2-2〉 분산 시스템 SW 미들웨어 분류

분류	설명
웹 애플리케이션 서버	<ul style="list-style-type: none"> <li>- Web Application Server</li> <li>- 웹 시스템에서 전달된 Request를 처리하기 위해 트랜잭션 관리, 세션 유지, 부하 분산 등의 역할을 하는 서버의 소프트웨어</li> </ul>
연계 통합 솔루션	<ul style="list-style-type: none"> <li>- 시스템 간 표준화된 데이터 송수신 처리를 통해 통합 환경 구성 지원</li> <li>- EAI(Enterprise Application Integration, 각 애플리케이션에 어댑터를 설치하고 중앙 허브를 통해 연결하는 방식), ESB(Enterprise Service Bus, 시스템 간 연계 및 메시징 변환, 라우팅 등의 작업을 표준 기반 인터페이스를 통해 연결하는 방식) 등 솔루션 존재</li> </ul>
실시간 데이터 처리	<ul style="list-style-type: none"> <li>- 지속적으로 발생하는 데이터를 실시간 분석하고 반응하는 시스템</li> <li>- CEP(Complex Event Processing, 실시간 발생하는 다양한 데이터를 조합하여 복잡한 상황에 대한 이벤트 또는 유형을 추론하는 방식) 등 존재</li> </ul>
분산 병렬 처리	<ul style="list-style-type: none"> <li>- 대규모 데이터를 실시간 처리, 분석하기 위해 다수의 노드에서 분할 처리</li> <li>- DDS(Data Distribution Service, 각 시스템에서 교환 데이터 유형을 정의하면 해당 데이터 교환 대상 시스템 검색, 교환 등 처리 수행) 등 존재</li> </ul>
TP 모니터	<ul style="list-style-type: none"> <li>- Transaction Processing Monitor</li> <li>- 트랜잭션 처리 모니터링 및 제어 시스템</li> </ul>

#### (2) IT 자원 관리

〈표 2-3〉 IT 자원 관리 미들웨어 분류

분류	설명
시스템 관리	<ul style="list-style-type: none"> <li>- 서버의 리소스 및 프로세스 관리 시스템</li> <li>- 하드웨어 자원 관리, 서버 가용성 모니터링 및 측정, SW 자원 관리, 사용자별 권한 및 사용현황 관리, 서버 보안 관리 등 수행</li> </ul>
SW 실행 관리	<ul style="list-style-type: none"> <li>- 시스템 소프트웨어의 실행 상태 관리 시스템</li> <li>- 실행 소프트웨어의 아키텍처 분석, 사용자 체감 성능 모니터링, 트랜잭션 자동 분석 및 보고서 생성 등 수행</li> </ul>
네트워크 관리	<ul style="list-style-type: none"> <li>- 네트워크 장비, 회선, 트래픽 등 관리 시스템</li> <li>- 네트워크 장애 검출 및 대응, 네트워크 구성 관리, 네트워크 성능 측정 및 분석/보고, 네트워크 보안 예방/탐지/억제/복구 관리 등 수행</li> </ul>
IT 서비스 운영 관리	<ul style="list-style-type: none"> <li>- IT 시스템의 운영 과정 모니터링 및 관리 시스템</li> <li>- 헬프 데스크(Help Desk), 구성 관리, 형상 관리, 문제 관리, 장애 관리, 이행 관리 등 수행</li> </ul>

### (3) 서비스 플랫폼

〈표 2-4〉 IT 자원 관리 미들웨어 분류

분류	설명
IoT 플랫폼	- IoT 장치를 연결하고 응용 서비스의 설치, 구동, 정지, 해제 등을 제어하고 관리하는 플랫폼
클라우드 서비스 플랫폼	- 클라우드 서버 기반 서비스 제공 플랫폼
UI/UX 프레임워크	- 사용자와 소프트웨어 간 소통 기능 구성을 위해 필요한 라이브러리 및 응용소프트웨어 집합
CDN	- Content Delivery Network - 다수의 노드에 콘텐츠를 복제 저장하여 사용자가 인접 노드에서 빠르게 콘텐츠를 받을 수 있도록 지원하는 시스템

### (4) 네트워크 보안

〈표 2-5〉 네트워크 보안 미들웨어 분류

분류	설명
네트워크 접근 제어	- 인가된 사용자만 네트워크에 접근할 수 있도록 검사 및 차단 관리 기술
보안 통신	- 네트워크의 모든 통신 데이터를 암호화하여 데이터의 유출 및 변조를 방지하는 기술
침입 방지/사고 대응	- 네트워크의 비정상적 트래픽을 탐지하고 대응하는 기술 - SIEM(Security Information and Event Management) 솔루션 등
보안 관리	- 다양한 보안 정책을 통합 관리하고 적용 현황을 분석하여 보안위협을 탐지하고 관리하는 시스템

## 수행 내용 / 미들웨어 파악하기

---

### 재료 · 자료

- 시스템 구성도
- 요구사항 정의서

### 기기(장비 · 공구)

- 오픈소스 기반 미들웨어
- 자료 조사를 위한 인터넷 검색 환경
- 문서 작성 도구, Presentation Tools, 뷔프로젝터

### 안전 · 유의 사항

- 다양한 사례 조사를 수행하고 적극적으로 조사 결과를 공유한다.
- 사례 분석을 통해 이해한 내용은 토론을 통해 이해도를 점검한다.

### 수행 순서

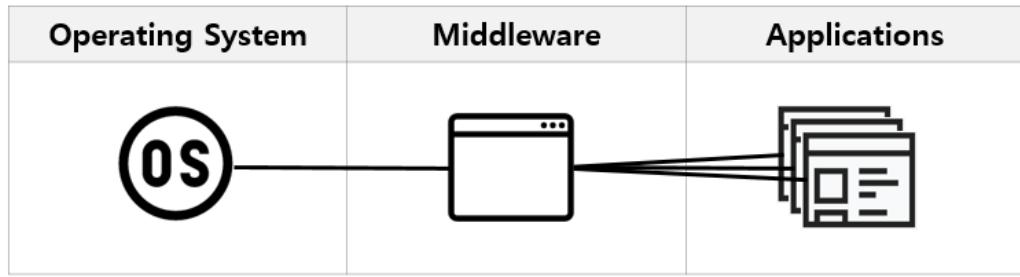
#### ① 운영체제와 응용소프트웨어 사이에 존재하는 미들웨어의 역할을 파악한다.

운영체제는 응용소프트웨어를 실행시키고 정상적으로 구동될 수 있도록 관리하며 응용소프트웨어 간에 통신을 지원하는 역할을 수행한다. 이러한 처리는 단일 시스템 내에서 수행되는 경우도 있지만 물리적으로 분리되어 있는 다수의 시스템 간에도 수행되기 때문에, 운영체제는 복잡하게 얹혀 있는 응용소프트웨어의 프로세스와 자원들을 통제해야 한다. 미들웨어는 운영체제와 응용소프트웨어 사이에 위치하며 복잡한 처리가 원활하게 수행되도록 지원한다. 미들웨어의 개념과 필요성을 학습하고 미들웨어의 역할을 파악해 본다.

##### 1. 미들웨어의 개념을 학습하고 필요성을 분석한다.

###### (1) 미들웨어의 개념을 학습한다.

미들웨어의 개념을 이해하기 위해 학습 내용 2-1의 필요 지식 ①-1을 참고하여 학습한다. 사용자는 미들웨어의 존재를 인지할 필요는 없지만 복잡한 시스템 구성일수록 미들웨어는 중요한 역할을 수행하기 때문에 응용소프트웨어 엔지니어는 정확히 알고 활용할 수 있어야 한다.



출처: 집필진 제작(2021)  
[그림 2-1] 미들웨어 개념도

## (2) 미들웨어의 필요성을 분석한다.

시스템의 규모가 점점 커지고 시스템 간 연결 관계 및 시스템 구성이 복잡해지면서 미들웨어의 필요성은 점점 부각되고 있다. 운영체제와 응용소프트웨어 입장에서 미들웨어가 필요한 이유를 생각해보고 그룹 토의를 통해 공유하여 정리한다.

〈표 2-6〉 미들웨어 필요성 정리 예시

구분	필요성	설명
운영체제	처리 효율 향상	<ul style="list-style-type: none"> <li>- 다수의 응용소프트웨어가 개별적으로 운영체제에 요청하는 작업을 미들웨어가 중간에서 통합하고 정리하여 운영체제에 요청함으로써 작업 효율 향상</li> </ul>
	유지보수성 향상	<ul style="list-style-type: none"> <li>- 개선 요구사항 발생 시 연결된 모든 응용소프트웨어를 고려할 필요 없이 미들웨어만 고려함으로써 개선 작업이 용이함</li> </ul>
응용 소프트웨어	개발 생산성 향상	<ul style="list-style-type: none"> <li>- 시스템 간 상호작용을 위해서 응용소프트웨어마다 개발해야 하는 부분을 별도로 분리하여 미들웨어로 개발하고, 각각의 응용소프트웨어에서는 미들웨어를 활용하여 처리함으로써 중복 개발 최소화</li> </ul>
	안정성 향상	<ul style="list-style-type: none"> <li>- 시스템 제어를 위한 복잡한 기능을 응용소프트웨어마다 직접 개발하는 대신 검증된 미들웨어 모듈을 활용함으로써 시스템 전반의 안정성 향상</li> </ul>

## 2. 미들웨어의 역할을 파악한다.

### (1) 미들웨어의 주요 기능을 조사한다.

미들웨어로 구현할 수 있는 기능들을 학습 내용 2-1의 필요 지식 ①-2를 참고하여 조사하고 정리한다.

### (2) 미들웨어의 유형을 조사한다.

미들웨어는 기능에 따라 다양한 형태로 구성 가능하므로 요구사항에 따라 적절한 유형을 선택하여 적용할 수 있어야 한다. 학습 내용 2-1의 필요 지식 ①-3을 참고하여 미들웨어의 유형을 조사하여 정리한다.

### (3) 미들웨어 적용 사례를 조사한다.

미들웨어에 대한 이해를 돋기 위하여 유형별로 실무에 적용한 사례들과 적용 솔루션들을 조사하고 정리한다.

〈표 2-7〉 미들웨어 적용 사례

분류	적용 사례	설명
웹 애플리케이션 서버	쇼핑몰 실행환경 구성	- 웹 기반 쇼핑몰 구축 시 오픈소스 웹 애플리케이션 서버인 아파치 톰캣(Apache Tomcat)을 이용하여 JSP, Java Servlet 실행환경 구성
연계 통합 솔루션	회계 시스템과 쇼핑몰 주문 실적 데이터 인터페이스	- 시스템 간 인터페이스 개발 시 오픈소스 통합 프레임워크인 아파치 카멜(Apache Camel) 활용
분산 병렬 처리	쇼핑몰 판매 현황 실시간 분석	- 인메모리(In-Memory) 기반 오픈소스 분산 병렬 처리 솔루션인 아파치 스파크(Apache Spark)를 이용해 주문 현황의 실시간 분석 기능 구현
TP 모니터	쇼핑몰 성능 모니터링	- 쇼핑몰 장애 예방을 위해 오픈소스 모니터링 도구인 Scouter를 이용해 웹 애플리케이션 서버 모니터링

### (4) 미들웨어의 역할을 파악한다.

위에서 조사한 미들웨어의 주요 기능 및 유형별 실무 적용 사례들을 학습 그룹 구성원들과 공유하고 토론을 통해 미들웨어의 역할을 파악하여 정리한다.

〈표 2-8〉 미들웨어 역할 정리 예시

역할	설명	주요 오픈소스 SW
통합 관리	- 물리적으로 분리되거나 이기종인 시스템들을 통합하여 중앙에서 모니터링하고 관리 - 웹 애플리케이션 서버, SW 실행 관리 등 활용	- Apache Tomcat - Prometheus - Scouter
시스템 간 연계	- 물리적으로 분리되거나 이기종인 시스템들 사이에서 데이터를 인터페이스하고 워크플로 연결 - 연계 통합 솔루션 등 활용	- Apache Camel - Apache ServiceMix - Apache ActiveMQ
보안 관리	- 응용소프트웨어 활용을 위해 적용되어야 하는 기본적인 보안 정책 관리 및 보안 통제 관리 - 네트워크 접근 제어, 보안 통신, 침입 탐지 등 활용	- pfSense - Snort IPS - Squid
개발 지원	- 시스템 개발을 위해 기본적으로 필요한 골조 제공 - IoT 플랫폼, UI/UX 프레임워크 등 활용	- Mobius - bootstrap

## ② 미들웨어에서 제공하는 기능 및 처리 흐름을 파악한다.

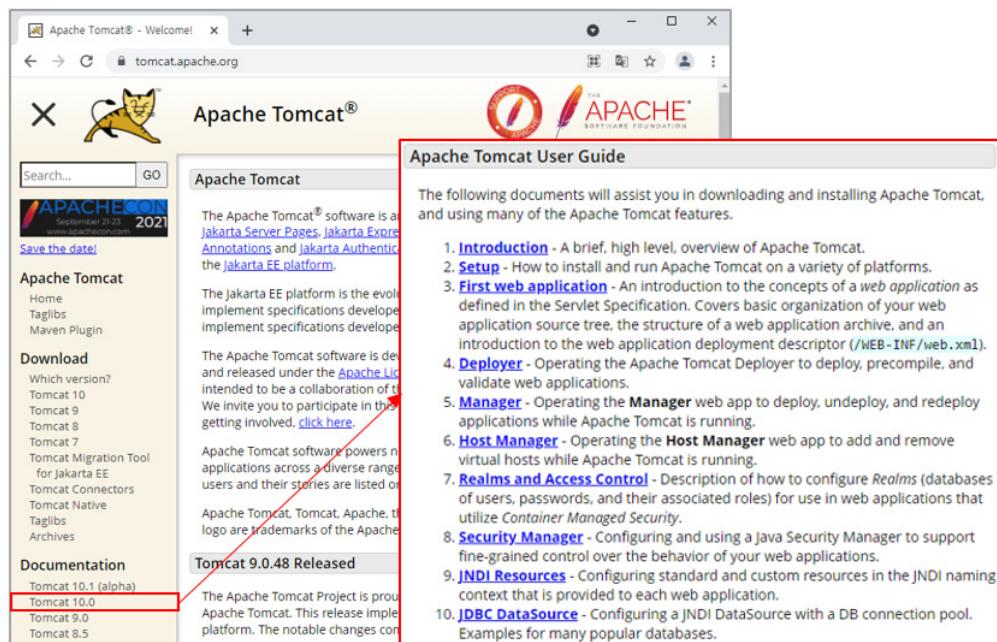
앞에서 학습한 것과 같이 미들웨어가 수행할 수 있는 역할이 다양하며 적용 가능한 유형도 다양하다. 그중에서 응용소프트웨어 엔지니어링을 위해 가장 기본적으로 활용하는 웹 애플리케이션 서버를 중심으로 미들웨어가 제공하는 세부 기능과 처리 흐름을 파악한다.

### 1. 미들웨어에서 제공하는 기능을 파악한다.

오픈소스 웹 애플리케이션 서버 중에서 가장 널리 사용되는 아파치 톰캣(Apache Tomcat)을 중심으로 미들웨어가 제공하는 기능을 파악한다.

(1) 아파치 톰캣 홈페이지 <https://tomcat.apache.org/>에 접속한다.

(2) Documentation 메뉴를 클릭하여 User Guide를 확인한다.



출처: 임필진 제작(2021)  
[그림 2-2] 아파치 톰캣 기능 확인 방법

(3) 웹 애플리케이션 서버에서 제공 가능한 기능을 확인하여 정리한다.

〈표 2-9〉 웹 애플리케이션 서버 주요 기능 정리 예시

구분	주요기능	설명
자원 관리	- DB 커넥션	- DB 커넥션 풀 및 JNDI 데이터 소스 설정
	- 모니터링 관리	- 아파치 톰캣 모니터링 및 관리 도구 사용
보안 관리	- SSL 지원	- https 프로토콜 사용을 위한 SSL 인증서 설정
	- 프록시 지원	- 프록시 서버 뒤에서 실행되도록 환경 설정

### 수행 tip

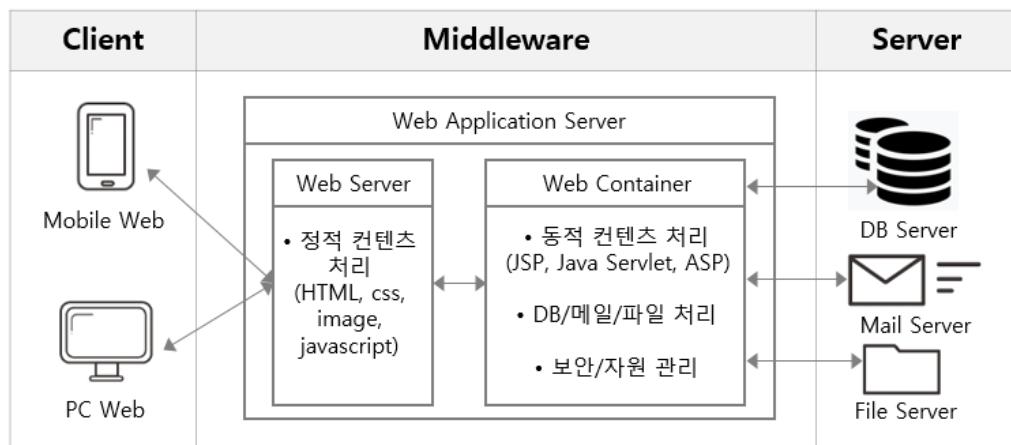
- 웹 애플리케이션 서버 이외에 다른 미들웨어들도 유사한 방식으로 제공 기능 및 처리 흐름을 파악할 수 있다. 그 룹 토의 및 실습으로 타 미들웨어도 파악해본다.
- 아파치 톰캣과 같은 오픈소스 소프트웨어의 경우 모든 소스코드가 공개되어 있으므로 다운받아서 직접 수정하여 사용할 수 있으며, 오픈소스 프로젝트에 직접 참여하거나 커뮤니티에 참여하여 정보를 공유할 수도 있다.

## 2. 미들웨어에서 제공하는 처리 흐름을 파악한다.

웹 애플리케이션 서버의 구성을 확인하고 처리 흐름을 파악한다.

### (1) 웹 애플리케이션 서버의 구성을 확인한다.

[그림 2-3]을 참고하여 웹 애플리케이션 서버의 구성을 확인한다. 웹 애플리케이션 서버는 웹 서버(Web Server)와 웹 컨테이너(Web Container)로 구성되며 웹 서버는 정적 콘텐츠 처리, 웹 애플리케이션 서버는 동적 콘텐츠 처리를 수행한다.



출처: 집필진 제작(2021)

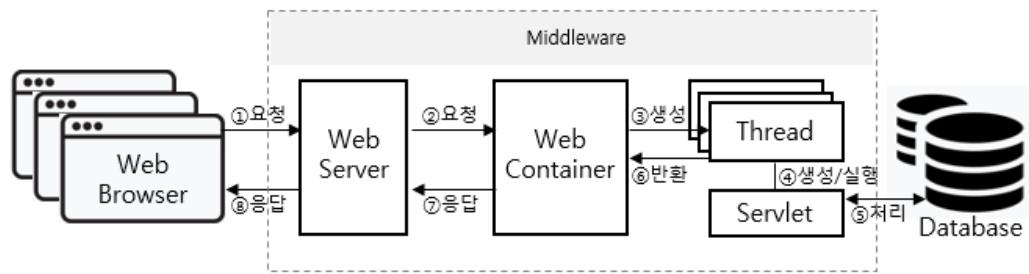
[그림 2-3] 웹 애플리케이션 서버 구조

### 웹 서버(Web Server)와 웹 애플리케이션 서버(Web Application Server)

웹 서버는 위의 [그림 2-3]과 같이 웹 애플리케이션 서버에 포함하여 구성 가능하나 일반적으로는 웹 애플리케이션 서버 앞에 별도로 분리하여 구성한다. 보안을 고려하여 웹 애플리케이션 서버의 중요한 설정 정보가 외부에 노출되는 위험을 최소화하기 위해 DMZ(DeMilitarized Zone, 내외부 네트워크 경계 지역) 구간에 웹 서버만 분리하여 설치하기도 하고, 서비스 안정성을 고려하여 웹 애플리케이션 서버가 다운되어도 웹 서버가 사용자에게 안내 메시지를 보여줄 수 있도록 분리하기도 한다. 또한, 서비스 성능을 고려하여 웹 서버가 처리 가능한 요청은 웹 애플리케이션 서버까지 전달하지 않고 바로 응답함으로써 웹 애플리케이션 서버의 부하를 경감시키기 위한 목적으로 분리하기도 한다.

## (2) 웹 애플리케이션 서버의 처리 흐름을 파악한다.

[그림 2-4]은 웹 애플리케이션 서버를 이용해 데이터베이스를 조회하는 웹 시스템의 처리 흐름 예시이다. [그림 2-4]를 참고하여 웹 애플리케이션 서버의 처리 흐름을 파악한다.



출처: 집필진 제작(2021)

[그림 2-4] 웹 애플리케이션 서버 처리 흐름 예시

(가) 웹 브라우저에서 웹 서버로 데이터 조회(Request) 처리를 확인한다.

웹 브라우저에서 HttpRequest를 전송한다.

(나) 웹 서버가 동적 콘텐츠 제공이 불가능하여 웹 컨테이너로 조회 요청을 전달하는 처리를 확인한다.

웹 서버는 HttpRequest를 웹 컨테이너로 전달한다.

(다) 웹 컨테이너가 스레드(Thread)를 생성하여 데이터베이스와 커넥션을 맺고 데이터를 조회하는 서블릿(Servlet)을 실행하는 과정을 확인한다. 이때 컨테이너에 서블릿이 생성되지 않은 경우 서블릿을 먼저 생성 및 초기화하고 실행하는 것을 확인한다.

웹 컨테이너는 효율적으로 자원을 관리하기 위하여 서블릿을 컨테이너당 한 개만 생성하여 관리하며 스레드를 스레드 풀(Thread Pool)을 이용해 적정량 만 생성하여 관리한다. 웹 서버에서 전달받은 HttpServletRequest 객체는 서블릿이 이해할 수 있도록 HttpServletRequest로 변환하고 HttpServletResponse를 생성하여 서블릿에 전달한다.

(라) 서블릿이 데이터베이스를 조회하고 결과를 컨테이너에 전달하는 처리를 확인한다.

서블릿은 HttpServletRequest의 요청 내용을 이용하여 데이터베이스를 조회하고, 조회 결과는 HttpServletResponse에 담아서 웹 컨테이너에 전송한다.

(마) 웹 컨테이너가 처리 결과를 웹 서버에 전송하는 과정을 확인한다.

웹 컨테이너는 HttpServletResponse를 HttpResponse로 변환하여 웹 서버에 전송한다.

(바) 웹 서버가 웹 브라우저로 처리 결과를 전달하는 과정을 확인한다.

웹 서버는 HttpResponse를 웹 브라우저로 전달한다.

## 2-2. 미들웨어 운용

### 학습 목표

- 응용소프트웨어 특성에 따라 선정된 미들웨어를 운용할 수 있다.

### 필요 지식 /

#### ① 전자정부 표준프레임워크

##### 1. 전자정부 표준프레임워크의 개념

정부 부처, 지자체, 공공기관 등의 공공정보화사업에서 JAVA 기반 웹/모바일 시스템 구축 시 활용되는 개발프레임워크로 한국지능정보사회진흥원(NIA)의 표준프레임워크 센터에서 Apache 2.0 라이선스로 공개하고 있어 일반 기업 및 학습용으로도 활용 가능하다.



출처: 집필진 제작(2021)

[그림 2-5] 전자정부 표준프레임워크 특징

##### 2. 전자정부 표준프레임워크의 구성

전자정부 표준프레임워크는 로그인, 세션 관리 등 웹/모바일 시스템 구축 시 공통적으로 개발하는 부분을 공통 컴포넌트로 제공하고 개발에 필요한 오픈소스 기반의 개발 도구, 실행환경, 관리환경, 운영환경 등을 표준 프레임워크로 제공한다.



출처: 집필진 제작(2021)  
[그림 2-6] 전자정부 표준프레임워크 구성

〈표 2-10〉 전자정부 표준프레임워크 구성 요소의 기능 및 역할

구분	기능 및 역할
실행환경	- 화면, 서버 프로그램, 데이터 개발, 배치 프로그램의 개발 표준 지원
개발환경	- 데이터 처리, 테스트 자동화, 코드 검사 등 개발에 필요한 오픈소스 SW 지원
운영환경	- 서비스 운영을 위한 시스템 설정, 모니터링, 배치 실행 및 스케줄링 지원
관리환경	- 개발 결과물의 안정적인 시스템 배포 및 운영 관리 지원
공통컴포넌트	- 사용자 관리, 인증, 게시판 등 공통 기능으로 재사용 가능한 컴포넌트 제공

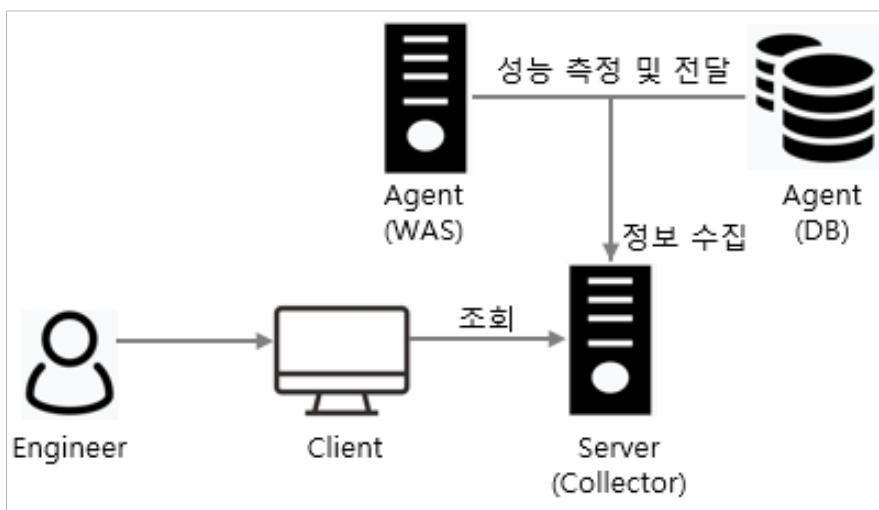
## ② Scouter

### 1. Scouter 개념

Scouter는 오픈소스 소프트웨어(Apache-2.0 License)로 공개된 애플리케이션 성능 관리(APM: Application Performance Management) 도구로서 응용소프트웨어 엔지니어가 분산 환경에서 안정적으로 시스템을 운영 및 관리하도록 지원하는 미들웨어이다. C/S(Client/Server) 형태가 일반적이지만 플러그인을 이용하면 Web 기반으로도 이용이 가능하며 오픈소스 소프트웨어이므로 학습용으로도 적합하다.

## 2. Scouter 구성

Scouter는 Server, Client, Agent로 구성된다.



출처: 집필진 제작(2021)

[그림 2-7] Scouter 구성도

〈표 2-11〉 Scouter 구성 요소

구분	설명
Agent	- WAS(Web Application Server), Database 등 모니터링 대상 시스템들에 설치하여 성능을 측정하고 그 결과를 Server로 전송
Server	- 각 Agent에서 측정된 결과 수집 및 저장
Client	- 서버 요청 및 응답 현황, 응답 평균속도, CPU 사용량, JVM 메모리 사용량, GC 시간 등 Server에 수집된 정보 확인

### 수행 tip

- Scouter 이외에도 Prometheus, Sensu 등 다양한 오픈소스 APM(Application Performance Management) 도구가 존재한다. 다른 도구들도 활용해보고 본인에 적합한 도구를 찾아본다.

## 수행 내용 / 미들웨어 운용하기

---

### 재료 · 자료

- 응용소프트웨어 구축 계획서
- 응용소프트웨어 개발 표준 정의서
- 요구사항 정의서

### 기기(장비 · 공구)

- 개발 프레임워크
- Application Performance Monitoring 소프트웨어
- Development Kit 소프트웨어

### 안전 · 유의 사항

- 학습자의 역량을 고려하여 실습 대상 미들웨어를 선정한다.
- 개발자 환경 구축(2001020233\_19v4) 학습모듈의 선행 학습을 권장한다.
- 오픈소스 소프트웨어의 버전은 지속적으로 변경되므로 실습 시 본 교재와 다른 버전을 이용하게 되는 경우 소프트웨어 간 호환성을 고려하여 버전을 선택해야 실습이 용이하다.

### 수행 순서

#### ① 응용소프트웨어 특성에 따라 미들웨어를 선정하여 설치한다.

미들웨어 운용을 실습하기 위해 기업 내에서 활용 가능한 웹 시스템 구축을 가정하여 적합한 미들웨어를 선정하고 설치하여 학습 환경을 구성한다.

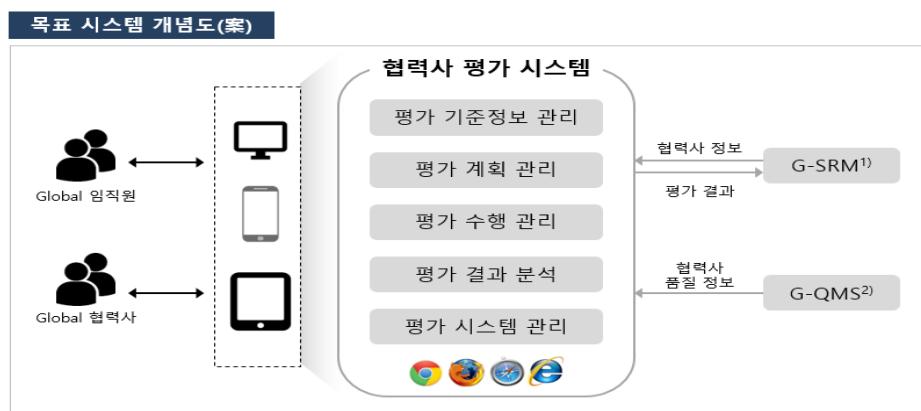
##### 1. 응용소프트웨어 특성을 파악한다.

###### (1) 응용소프트웨어 구축 계획서 내용을 파악한다.

아래 [그림 2-8], [그림 2-9], [그림 2-10]의 예시를 참고하여 응용소프트웨어의 구축 개요 및 요구 기능 등을 파악한다.

프로젝트 명/기간
1. 프로젝트 명 : 협력사 평가 시스템 구축 PJT
2. 프로젝트 수행 기간 : '21.04 ~ '21.10 (착수 시점 기간 내 협의 예정, Open Target : '21.11)
배경 및 목적
1. 배경 : 비대면 평가 요구 증가 / Global 평가 눈높이 동질화 필요성 증가 / 평가 소요 공수 감소 필요 2. 목적 : 모바일 시스템 활용한 평가로 평가 업무 효율화, Global 평가 신뢰성 확보, 신규 사업 기회 확대
목표 시스템 설명
1. 협력사 평가 계획 수립 및 평가 수행 관리 시스템 - 협력사 정보, 평가 지표, 평가 프로세스, 평가 산식, 평가 점수표 등 기준정보 관리 - 평가 유형별 표준 프로세스 구축 및 협력사 환경別 평가 프로세스 조정 - PC, 태블릿, 휴대폰 등 다양한 Device 통한 현장 평가 결과 입력 및 평가 점수 자동 산출 2. Global 확산을 고려한 시스템으로 다국어 적용 - 확산 예정 지역 : 한국, 중국, 인도, 미국, 유럽, 멕시코, 브라질 - 적용 예정 언어 : 한국어, 중국어, 영어

[그림 2-8] 응용소프트웨어 구축 계획서의 구축 개요 예시



2) G-QMS : Global Quality Manage

구분	주요 기능	설명
시스템 관리	사용자 및 조직 관리	<ul style="list-style-type: none"> <li>Global 임직원 &amp; Global 협력사 사용자 관리</li> <li>SRM 시스템과 사용자 및 조직 정보 인터페이스</li> <li>사용자 정보 수기 등록 관리</li> </ul>
	화면 메뉴 및 권한 관리	<ul style="list-style-type: none"> <li>화면 메뉴 구성 관리</li> <li>Role Base 권한 관리 및 권한 별 메뉴 관리 (下記 예시 참고)             <ol style="list-style-type: none"> <li>System Administrator : 최상위 권한</li> <li>Global Master : Global 법인의 기준 정보 수정 &amp; 모든 권한</li> <li>Local Master : 소속 법인의 기준 정보 수정 &amp; 모든 권한</li> <li>General : 평가결과 입력, 조회, 분석 결과 조회</li> <li>Supplier : 소속 회사 정보 입력 및 조회 가능, 외부망 제한적 허용</li> </ol> </li> <li>사용자 휴직/퇴사/장기 미사용 등 상태에 따른 권한 자동 회수</li> </ul>
	계시판 및 공지 관리	<ul style="list-style-type: none"> <li>계시판 및 팝업 공지</li> </ul>
	공통 코드 관리	<ul style="list-style-type: none"> <li>공통 코드 관리, 정렬 순서 지정 관리</li> </ul>
기준 정보 관리	협력사 관리	<ul style="list-style-type: none"> <li>협력사 일반 정보 관리</li> <li>SRM 시스템과 협력사 정보 인터페이스</li> <li>품질 관리 시스템 (G-QMS)와 협력사 품질 정보 인터페이스</li> </ul>
	평가 기준 정보 관리	<ul style="list-style-type: none"> <li>각 평가 항목(최소요구사항 301개 항목) 별 관련 서류, 기준, 조항 등 관리</li> <li>평가 항목 별 중요도 관리</li> </ul>
	평가 요구 사항 관리	<ul style="list-style-type: none"> <li>평가 항목별 평가 질문 사항 등록             <ol style="list-style-type: none"> <li>각 평가 항목별 요구사항 관리</li> <li>각 평가 항목별 통수</li> <li>평가 요구 사항 등록</li> </ol> </li> </ul>

출처: 집필진 제작(2021)  
[그림 2-10] 유플러스프트웨어 그룹 계회서이 그룹 가는 예시

## (2) 응용소프트웨어 특성을 파악한다.

예시 자료는 글로벌 내부 직원 및 외부 협력사가 평가 업무에 활용하기 위한 시스템으로 PC, 태블릿, 휴대폰에서 활용 가능한 시스템이다. PC와 모바일에서 모두 활용 가능한 웹 시스템으로 구성할 수도 있고 PC용 웹과 모바일용 앱을 별도로 구성할 수도 있기 때문에 요구 기능을 정확하게 파악해야 하며, 네트워크 상황이 열악한 해외사업장에서도 원활하게 시스템을 사용할 수 있도록 성능 측면의 특성을 파악해야 한다. 그룹 토의를 통해 다양한 시각으로 응용소프트웨어의 특성을 파악하여 정리한다.

## 2. 미들웨어를 선정한다.

파악된 응용소프트웨어의 특성을 고려하여 미들웨어를 선정한다. 웹 시스템을 구성하기 위해 웹 애플리케이션 서버를 선정하고 개발 생산성 향상을 위한 개발 프레임워크, 시스템 간 데이터 연계를 위한 EAI 도구, 성능 관리를 위한 모니터링 도구 등을 선정할 수 있으며, 실무에서는 기업마다 개발 표준을 사전에 정의해두고 이를 기반으로 선정하기도 한다. 그룹 토의를 통해 미들웨어를 선정하여 정리한다.

## 3. 미들웨어를 설치한다.

다양한 미들웨어 선정이 가능하나 이번에는 실습 환경을 고려하여 오픈소스 소프트웨어 기반으로 웹 시스템을 구축하는 환경을 정의하고 전자정부 표준프레임워크, Scouter 등 관련 프로그램을 설치한다. 전자정부 표준프레임워크와 Scouter에 대해서는 필요 지식을 참고하여 학습한다.

### (1) 웹 시스템 개발 및 실행환경을 구성하기 위한 미들웨어를 정의한다.

학습용 웹 시스템 구축을 위한 오픈소스 미들웨어를 정의한다.

〈표 2-12〉 웹 시스템 구축 환경 예시

구분	Software	License
개발 플랫폼	- openjdk-1.8.0.292-1.b10	- GNU GPL v2
데이터베이스	- PostgreSQL 9.4	- PostgreSQL (MIT 유사)
WAS	- Tomcat 8.5	- Apache License v2
개발 프레임워크	- eGovFrameDev-3.10.0-64bit	- Apache License v2
개발 도구	- Eclipse IDE 4.16	- Eclipse Public License v2
배포 도구	- Maven 3.9	- Apache License v2
APM	- Scouter 2.10.2	- Apache License v2

### 수행 tip

- 오픈소스 소프트웨어를 이용하는 경우에는 라이선스를 반드시 확인해야 한다. 특히 출원 예정인 소프트웨어를 GPL 라이선스 소프트웨어를 활용하여 만드는 경우 수정한 소스코드를 공개해야 해서 문제가 될 수 있다. 충분한 검토 후 이용하도록 한다.
- 응용소프트웨어 개발환경 구성 시에는 다양한 소프트웨어를 동시에 설치해야 하므로 소프트웨어들의 버전을 잘 확인해야 한다. 특히 전자정부 표준프레임워크의 경우 전자정부 표준프레임워크의 버전과 호환되는 각 소프트웨어의 버전을 제시하고 있으며 이를 준수하지 않는 경우 설치 및 실행에 많은 오류가 발생될 수 있다.

## (2) 응용소프트웨어 개발을 위한 미들웨어 소프트웨어를 설치한다.

아래 내용을 참고하여 순차적으로 미들웨어를 설치하고 환경 설정을 진행한다. 개발자 환경 구축 방법은 개발자 환경 구축(2001020233\_19v4) 학습모듈에서 상세히 다루기 때문에 이번 학습 내용에서는 필수 항목만 간단히 수행한다.

### (가) OpenJDK(Open Java Development Kit)를 설치한다.

OpenJDK 포털(<https://openjdk.java.net/>) 다운로드 메뉴에서 학습자의 운영체제에 맞는 소프트웨어를 다운로드 받아서 설치하고 시스템 환경변수에 JAVA\_HOME, Path를 설정한다. 정상적으로 설치되면 명령 처리기에서 java -version 명령어로 설치 버전을 확인할 수 있다.

### (나) PostgreSQL을 설치한다.

PostgreSQL 포털(<https://www.postgresql.org/>)의 다운로드 메뉴에서 학습자의 운영체제에 맞는 소프트웨어를 선택하여 설치한다. PostgreSQL의 pgAdmin으로 쿼리를 조작할 수도 있지만 DBeaver와 같은 데이터베이스 관리 도구를 이용하면 데이터베이스 조작이 용이하다.

### (다) 전자정부 표준프레임워크에서 제공하는 구현 도구를 설치한다.

전자정부 표준프레임워크 포털(<https://www.egovframe.go.kr/>)의 개발환경 다운로드 메뉴에서 학습자의 운영체제에 맞는 소프트웨어를 선택하고 포털의 개발 가이드를 참고하여 개발 도구인 이클립스(Eclipse)와 빌드 도구인 메이븐(Maven)을 설치한다.

### (라) Tomcat을 설치하고 구현 도구인 이클립스에 서버 설정을 한다.

Apache Tomcat 포털(<http://tomcat.apache.org/>)의 다운로드 메뉴에서 학습자의 운영체제에 맞는 소프트웨어를 선택하여 설치한 후 개발 가이드를 참

고하여 이클립스에 서버 설정을 한다.

(마) 전자정부 표준프레임워크에서 제공하는 공통 컴포넌트를 설치한다.

전자정부 표준프레임워크 포털(<https://www.egovframe.go.kr/>) 공통 컴포넌트 다운로드 메뉴의 배포 파일을 다운받고 개발 가이드를 참고하여 설치한다.

(바) 공통 컴포넌트의 실행을 위해 데이터베이스에 테이블을 생성하고 필수 데이터를 입력한다.

다운받은 공통 컴포넌트의 script 폴더에 포함되어 있는 PostgreSQL 테이블 생성용 DDL(Data Definition Language)과 데이터 생성용 DML(Data Manipulation Language)을 이용해 PostgreSQL에 테이블을 생성하고 필수 데이터를 입력한다.

(사) 공통 컴포넌트의 데이터베이스를 PostgreSQL로 설정한다.

전자정부 표준프레임워크는 다양한 DBMS(DataBase Management System)를 지원하며 설정을 바꿔서 사용할 수 있다. PostgreSQL을 이용할 수 있도록 환경을 설정한다.

#### 수행 tip

- 보안을 위해 DB 비밀번호는 암호화하여 관리해야 하지 만 학습용으로 전자정부 표준프레임워크를 간단하게 이용하는 경우에는 아래 [그림 2-12]를 참고하여 암호화 기능은 제외하는 것도 가능하다.

```
# 운영서버 타입(WINDOWS, UNIX)
Globals.OsType = WINDOWS
#
# DB서버 타입(mysql, oracle, altibase, tibero, cubrid, maria, postgres)
Globals.DbType = postgres
#
# 권한 인증방식(dummy, session, security) - 사용자의 로그인시 인증 방식을 결정함
# dummy : 딕미 방식으로 사용자 권한을 인증함
# session : 세션 방식으로 사용자 권한을 인증함
# security : spring security 방식으로 사용자 권한을 인증함
Globals.Auth = dummy
#
# MainPage Setting
Globals.MainPage = /EgovContent.do
#
#postgreSQL
Globals.postgres.DriverClassName=org.postgresql.Driver
Globals.postgres.Url=jdbc:postgresql://127.0.0.1:5432/postgres
Globals.postgres.UserName=postgres
Globals.postgres.Password=1234
```

출처: 집필진 제작(2021)

[그림 2-11] 전자정부 표준프레임워크 PostgreSQL DB 설정

The screenshot shows the Eclipse IDE interface with the file 'context-crypto.xml' open. The code defines a crypto configuration for 'egovCryptoConfig' with 'initial' set to 'false' and 'crypto' set to 'false'. It also specifies algorithm as 'SHA-256', algorithm key as 'egovframe', algorithm hash as 'gdyYs/IZqY86VclvhT8emCYfqY1ahw2vtLG+/FzNqtrQ=' (base64 encoded), and block size as '1024'. The 'cryptoPropertyLocation' is set to 'classpath:/egovframework/egovProps/globals.properties'.

```

<!--
  9   initial : globals.properties 연계 Url, UserName, Password 값 로드 여부(설정값 : true, false)
10  crypto : 계정 암호화 여부(설정값 : true, false)
11  algorith : 계정 암호화 알고리즘
12  algorithmKey : 계정 암호화 키
13  cryptoBlockSize : 계정 암호화키 블록사이즈
14  cryptoPropertyLocation : 설정파일 암복호화 경로 (선택) 기본값은 'classpath:/egovframework/egovProps/globals.properties'
-->
<egov-crypto:config id="egovCryptoConfig"
17    initial="false"
18    crypto="false"
19    algorithm="SHA-256"
20    algorithmKey="egovframe"
21    algorithmKeyHash="gdyYs/IZqY86VclvhT8emCYfqY1ahw2vtLG+/FzNqtrQ="
22    cryptoBlockSize="1024"
23    cryptoPropertyLocation="classpath:/egovframework/egovProps/globals.properties"
24  />

```

출처: 집필진 제작(2021)

[그림 2-12] 전자정부 표준프레임워크 DB Password 암호화 미사용 처리 설정

The screenshot shows the Eclipse IDE interface with the file 'context-datasource.xml' open. It defines a database connection 'dataSource' using 'BasicDataSource' with 'driverClassName' as 'org.apache.commons.dbcp2.BasicDataSource'. The 'url' is set to '\${Globals.postgres.Url}', 'username' to '\${Globals.postgres.UserName}', and 'password' to '\${egovEnvCryptoService.getPassword}' (highlighted by a red box).

```

18      <!-- DataSource -->
19      <alias name="dataSource" alias="egov.dataSource" />
20
21      <!-- PostresSQL -->
22      <beans profile="postgres">
23        <bean id="dataSource" class="org.apache.commons.dbcp2.BasicDataSource" dest...
24          <property name="driverClassName" value="${Globals.postgres.DriverClass...
25          <property name="url" value="${Globals.postgres.Url}" />
26          <property name="username" value="${Globals.postgres.UserName}" />
27          <!-- property name="password" value="#{egovEnvCryptoService.getPassword}...
28          <property name="password" value="${Globals.postgres.Password}" />
29        </bean>
30      </beans>

```

출처: 집필진 제작(2021)

[그림 2-13] 전자정부 표준프레임워크 암호화 전 DB Password 취득 처리 설정

#### (0f) 전자정부 표준프레임워크를 실행한다.

The screenshot shows the Eclipse IDE interface with the 'Console' tab selected, displaying Tomcat v8.5 Server logs. The logs show the startup of the server, including the start of the Coyote AbstractProtocol and Catalina, and a note about a 36496 ms startup duration.

```

Tomcat v8.5 Server at localhost [Apache Tomcat] C:\WeGovFrameDev-3.10.0-64bit\WeclipseWork...
2021-07-04 17:30:54,395  INFO [org.springframework.web.servlet.DispatcherServlet]
2021-07-04 17:30:54,395  INFO [org.springframework.web.servlet.DispatcherServlet]
2021-07-04 17:30:54,423 DEBUG [egovframework.rte.fdl.access.service.impl]
2021-07-04 17:30:54,437 DEBUG [egovframework.rte.fdl.access.service.impl]
2021-07-04 17:30:54,494  INFO [org.springframework.web.servlet.DispatcherServlet]
7월 04, 2021 5:30:54 오후 org.apache.coyote.AbstractProtocol start
정보: 프로토콜 핸들러 ["http-nio-8080"]을(를) 시작합니다.
7월 04, 2021 5:30:54 오후 org.apache.catalina.startup.Catalina start
정보: Server startup in 36496 ms
2021/07/04 17:30:54 [NONE] LoadJarBytes scouter.http 24016 bytes

```

출처: 집필진 제작(2021)

[그림 2-14] 전자정부 표준프레임워크 실행 성공 후 콘솔 출력 내용



출처: 집필진 제작(2021)

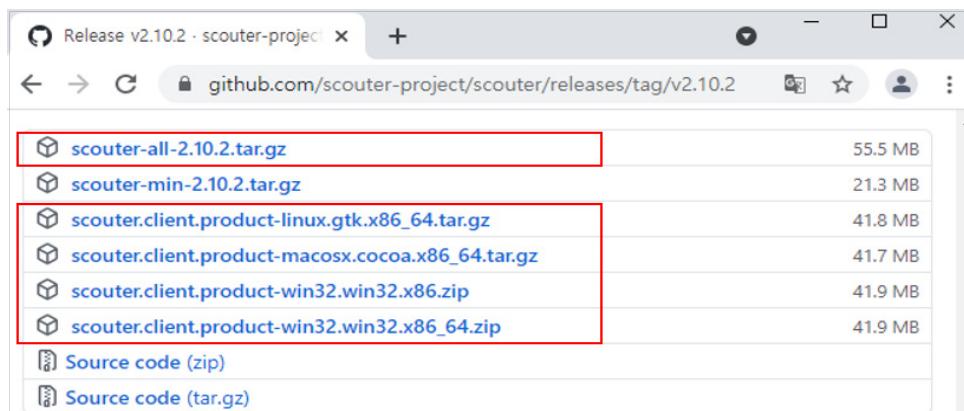
[그림 2-15] 전자정부 표준프레임워크 실행 성공 후 웹 시스템 접속 결과

### (3) 응용소프트웨어 운영을 위한 미들웨어 소프트웨어를 설치한다.

안정적인 운영을 위해 APM(Application Performance Monitoring) 도구인 Scouter를 설치한다.

#### (가) Scouter 설치 프로그램을 다운로드한다.

아래 [그림 2-16]을 참고하여 scouter 서버 및 에이전트 설치 파일과 클라이언트 설치 파일을 학습자의 운영체제에 맞춰서 다운로드받는다.

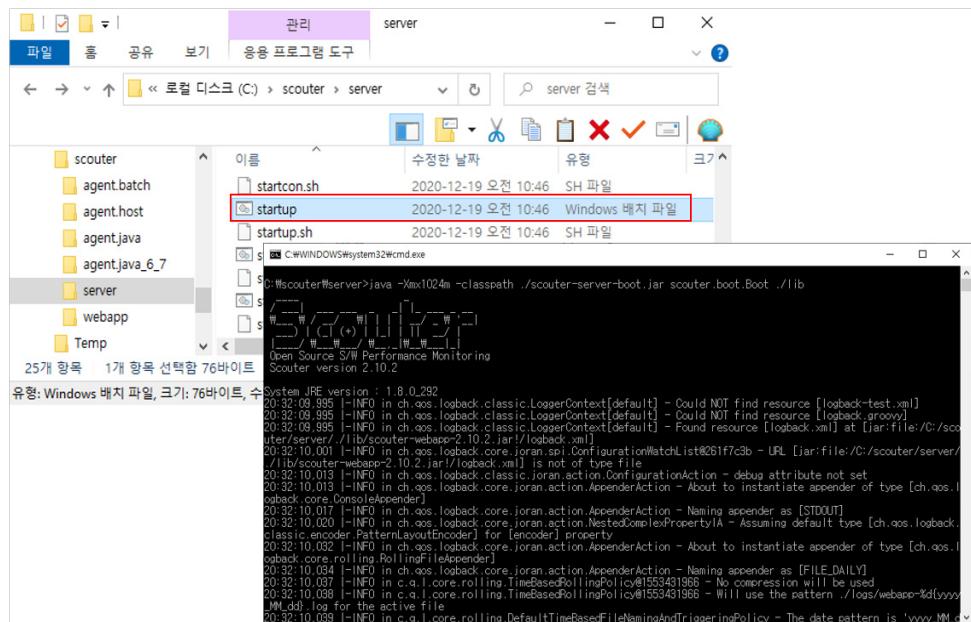


출처: 집필진 제작(2021)

[그림 2-16] Scouter 설치 파일 다운로드

#### (나) Scouter Server를 실행한다.

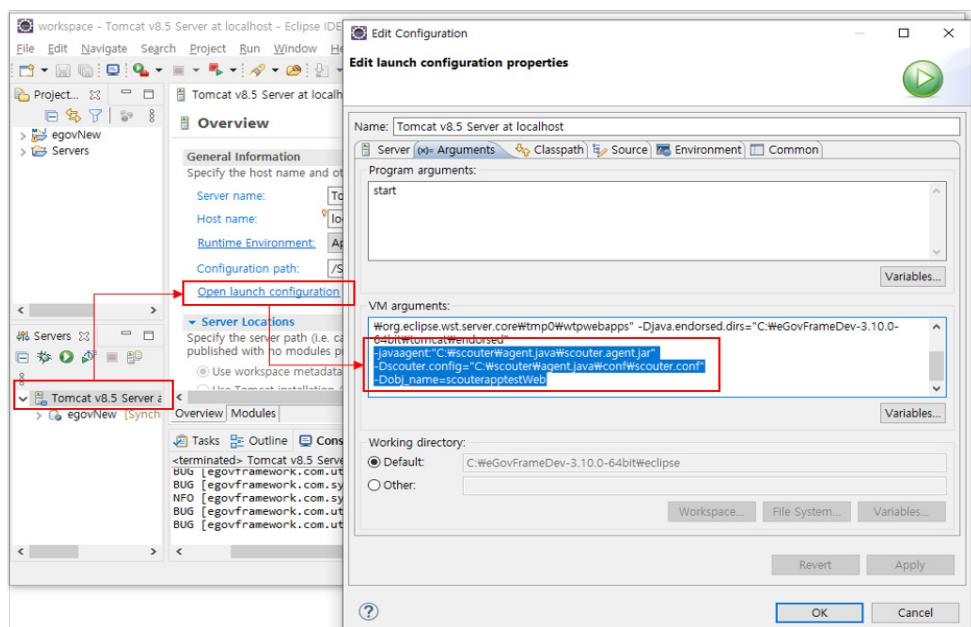
scouter-all을 C 드라이브에서 압축을 풀고 server 폴더 아래의 startup.bat 파일을 실행하여 Scouter Server를 기동한다. 명령 프롬프트에서 netstat -an:find 6100 명령어를 입력하면 Agent의 성능 측정 결과를 수신하기 위해 Listening 상태로 대기 중인 것을 확인할 수 있다.



출처: 집필진 제작(2021)  
[그림 2-17] Scouter Server 실행

#### (다) Scouter Agent를 실행한다.

WAS(Web Application Server)인 Tomcat의 성능을 측정하기 위해 이클립스의 서버 정보를 더블클릭하여 표시되는 Open Launch Configuration의 VM arguments에 아래 [그림 2-18]과 같이 Java Agent 정보를 추가하고 Tomcat을 실행한다.



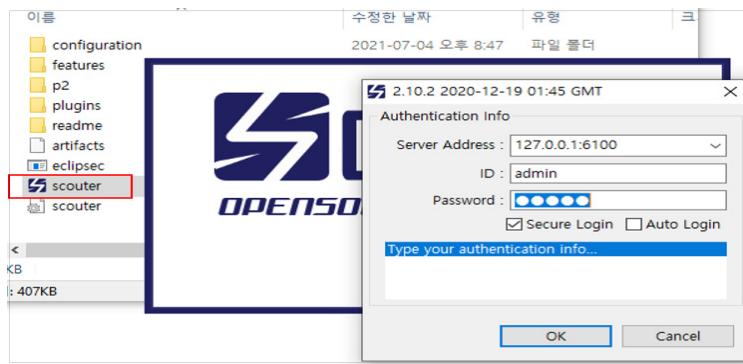
출처: 집필진 제작(2021)  
[그림 2-18] Scouter Java Agent 실행

### VM arguments

```
-javaagent:"C:\scouter\agent.java\scouter.agent.jar"  
-DScouter.config="C:\scouter\agent.java\conf\scouter.conf"  
-Dobj_name=scouterapptestWeb
```

#### (라) Scouter Client를 실행한다.

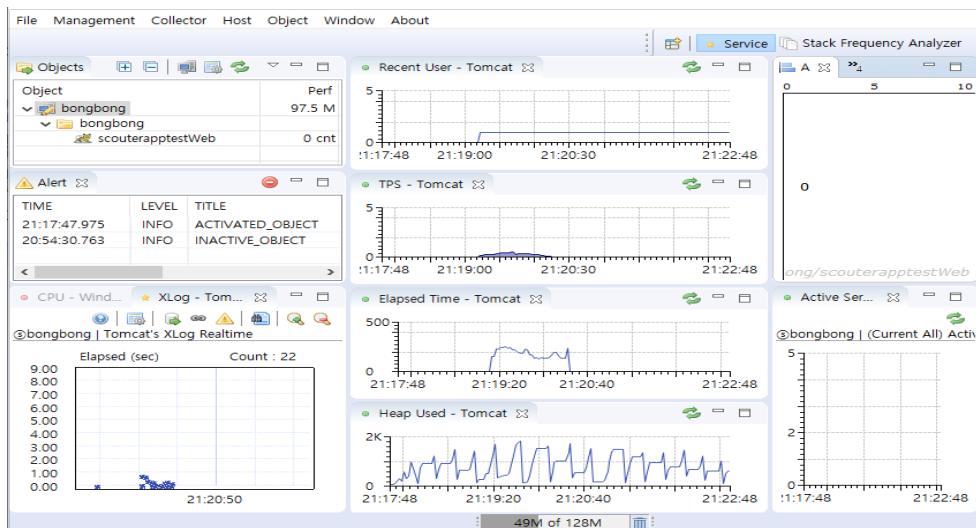
다운로드받은 scouter-client 파일의 압축을 풀어 scouter 파일을 실행하고 Server 정보와 ID, Password에 초기 기본값인 admin을 입력한다.



출처: 집필진 제작(2021)  
[그림 2-19] Scouter Client 실행

#### (마) WAS(Web Application Server) 성능을 모니터링한다.

scouter client를 통해 scouter server가 Java Agent로부터 수집한 Tomcat의 성능 측정 정보를 모니터링한다. Objects에서 Java Agent 설정에 입력했던 명칭을 확인할 수 있고 Recent User에서 최근 Tomcat 사용자 수를 확인할 수 있으며 TPS(Transaction Per Seconds) 통해 초당 트랜잭션 처리 건수를 확인할 수 있다.

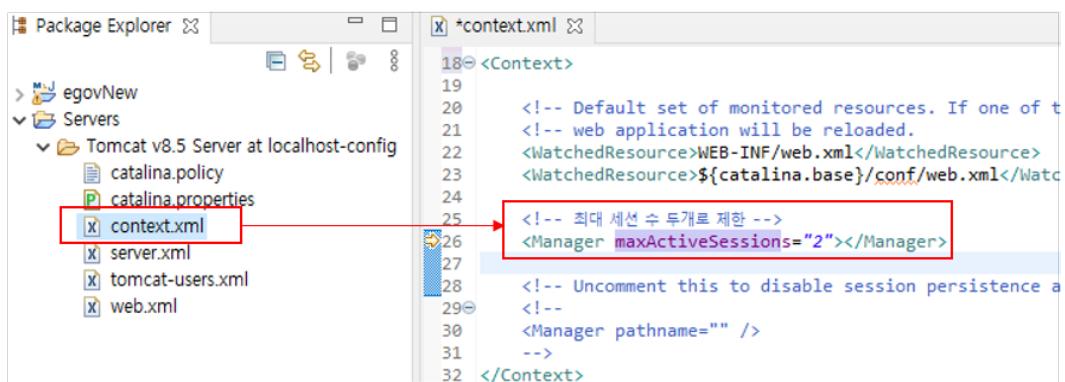


출처: 집필진 제작(2021)  
[그림 2-20] Scouter WAS 모니터링

## ② 응용소프트웨어 특성에 따라 선정된 미들웨어를 운용한다.

WAS(Web Application Server)에 접속 가능한 최대 세션 수가 초과하여 시스템을 이용할 수 없는 상황을 실습해보기 위해 Tomcat에 최대 사용자 수를 2로 변경한 후 3번 접속해보고 해당 시점에 Scouter의 변화를 확인한다.

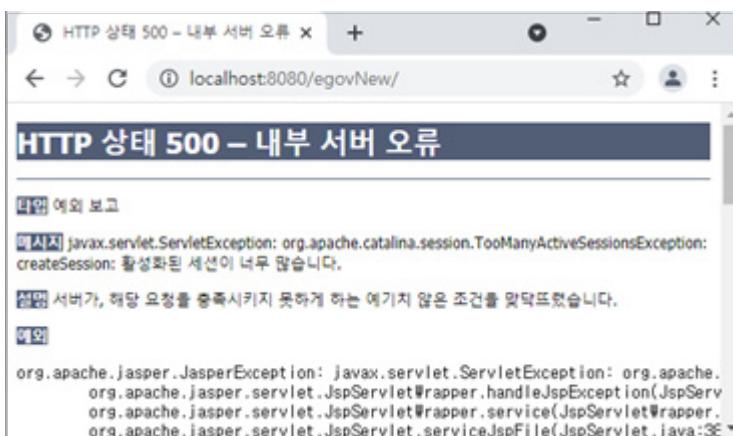
1. Tomcat의 최대 세션 수를 2로 설정하고 Tomcat을 재기동한다.



출처: 집필진 제작(2021)

[그림 2-21] Tomcat 최대 세션 수 설정

2. 시스템에 3회 접속을 시도하여 세션 초과 오류가 발생하는 것을 확인한다.



출처: 집필진 제작(2021)

[그림 2-22] Tomcat 세션 초과 오류 발생

### 수행 tip

- 동일한 브라우저에서 리프레시하는 것으로는 세션이 증가하지 않으므로 새로운 세션으로 창을 열어서 접속해야 한다.
- 실무에서는 세션 초과 오류를 포함하여 HTTP 오류가 바로 사용자에게 표시되지 않도록 에러 화면을 별도로 구현하여 사용자가 재시도하거나 시스템 담당자에게 문의해야 한다.

### 3. Scouter Client로 WAS 상태를 모니터링한다.

Recent User 수가 변동 없고 XLog에 Elapsed가 없이 빨간색 오류가 표시된다.



출처: 집필진 제작(2021)  
[그림 2-23] Scouter 세션 초과 모니터링

### 교수 방법

- 미들웨어 개념을 이해할 수 있도록 설명하고 다양한 유형의 미들웨어를 스스로 조사하며 미들웨어의 역할을 파악할 수 있도록 지도한다.
- 응용소프트웨어 유형에 따라 주로 사용하는 미들웨어 제품을 소개하고, 각 미들웨어에서 제공하는 기능 및 처리 흐름에 대해 사례 기반으로 이해하기 쉽게 설명한다.
- 실습 가능한 미들웨어를 선정하여 직접 설치하고 운용해볼 수 있도록 지도한다.
- 미들웨어 성능 관리에 대한 학습자의 지식 보유 수준을 확인하고 주요 사항을 간단히 설명한다.

### 학습 방법

- 미들웨어 개념을 학습하고 다양한 유형의 미들웨어를 조사하고 그룹 토의를 통해 조사 결과를 공유하여 이해도를 점검한다.
- 미들웨어의 제공 기능 및 처리 흐름을 학습하고 관련 미들웨어 제품들의 활용도를 분석하여 정리한다.
- 실습 가능한 미들웨어를 선정하여 직접 설치 및 운용해보고 문제가 발생하는 경우 그룹 스터디를 통해 해결한다.
- 미들웨어 성능 관리에 대해 학습하고 궁금한 사항은 질문한다.

## 학습2 평 가

### 평가 준거

- 평가자는 학습자가 학습 목표를 성공적으로 달성하였는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.

학습 내용	학습 목표	성취수준		
		상	중	하
미들웨어 파악	- 운영체제와 응용소프트웨어 사이에 존재하는 미들웨어의 역할을 파악할 수 있다.			
	- 미들웨어에서 제공하는 기능 및 처리 흐름을 파악할 수 있다.			
미들웨어 운용	- 응용소프트웨어 특성에 따라 선정된 미들웨어를 운용할 수 있다.			

### 평가 방법

- 서술형 시험

학습 내용	평가 항목	성취수준		
		상	중	하
미들웨어 파악	- 미들웨어 개념 이해 여부			
	- 미들웨어 분류 및 역할 파악 여부			
	- 미들웨어 기능 분석 능력			
	- 미들웨어 구성 및 처리 흐름 파악 여부			
미들웨어 운용	- 응용소프트웨어 특성에 따른 미들웨어 선정 능력			
	- 미들웨어 운용 도구 파악 능력			

• 작업장 평가

학습 내용	평가 항목	성취수준		
		상	중	하
미들웨어 운용	- 미들웨어 설치 및 운용 능력			
	- 미들웨어 모니터링 능력			

**피드백**

1. 서술형 시험

- 미들웨어의 개념을 정확히 이해했는지 평가하고, 미흡한 부분에 대해서는 사례를 이용해 부연 설명한다.
- 미들웨어의 분류 체계를 제대로 이해하고 미들웨어의 역할을 파악하고 있는지 평가하고, 미흡한 부분에 대해서는 모둠별 토의를 통해 이해도를 향상시키도록 지도한다.
- 미들웨어 기능에 대한 이해도를 평가하고 미흡한 경우 미들웨어 기능 분석 방법을 추가 설명하고, 학습자가 스스로 분석한 결과를 확인하여 이해 여부를 재점검한다.
- 미들웨어의 구성 및 처리 흐름 파악 여부를 확인하고, 성취수준이 낮은 학습자에게는 이해가 용이하도록 도식화하여 주요 사항을 설명한다.
- 응용소프트웨어 특성에 따른 미들웨어 선정 능력을 평가하고, 미흡할 때는 사례 조사자를 통해 보완한다.
- 미들웨어 운용에 필요한 도구들의 파악 능력을 평가하고, 평가 결과가 미흡한 경우에는 주요 사례를 기반으로 추가 설명한다.

2. 작업장 평가

- 미들웨어 설치 및 운용 능력을 평가하고, 미흡한 경우에는 범위를 축소하여 실습하도록 하고 성숙수준이 높은 학습자의 경우에는 모둠별로 미들웨어 고급 기능을 추가 학습하도록 한다.
- 미들웨어 모니터링 능력을 평가하고 미흡한 경우에는 모둠별로 결과 공유를 통해 보완한다.

학습 1	네트워크 기초 활용하기
학습 2	미들웨어 기초 활용하기

## 학습 3

## 데이터베이스 기초 활용하기

### 3-1. 데이터베이스 특징 식별

#### 학습 목표

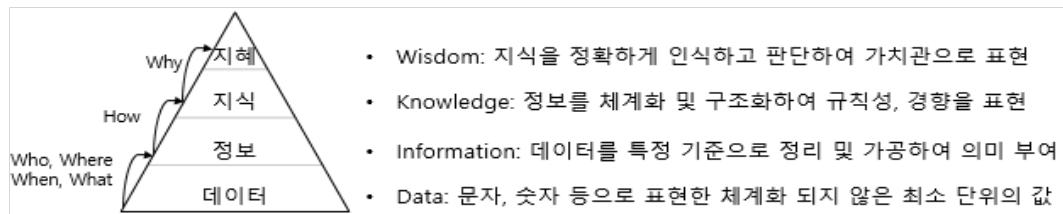
- 데이터베이스의 종류를 구분하고 응용소프트웨어 개발에 필요한 데이터베이스의 특징을 식별할 수 있다.

#### 필요 지식 /

##### ① 데이터베이스(Database)

###### 1. 데이터(Data)의 개념

데이터는 관찰이나 측정으로 수집한 사실(Fact)을 수치 또는 문자 형태로 표현한 최소 단위의 값이다. [그림 3-1]의 DIKW 피라미드와 같이 정보, 지식, 지혜로의 변환을 통한 창조(Creation)적 사고의 근간이 된다.



출처: 집필진 제작(2021)  
[그림 3-1] DIKW 피라미드

###### 2. 데이터베이스의 개념

데이터베이스는 공용으로 활용하기 위해 통합하여 저장한 운영 데이터의 집합이다.

〈표 3-1〉 데이터베이스 특성

특성	설명
실시간 접근성	- 요청받은 데이터 처리는 실시간으로 처리되고 결과를 반환해야 함
계속적 변화	- 저장된 데이터는 입력, 수정, 삭제에 의해 지속적으로 변화함
동시 공용	- 서로 다른 목적의 응용 SW 및 사용자에 의해 동시 공용 가능
내용에 의한 참조	- 데이터의 참조는 데이터의 주소가 아닌 저장된 값에 의해 처리됨

### 3. 데이터베이스 관리시스템(DBMS: Database Management System)의 개념

다수의 응용소프트웨어 및 사용자가 데이터베이스에 접근하여 원활하게 사용할 수 있도록 중간에서 관리해주는 시스템

〈표 3-2〉 데이터베이스 관리시스템 기능

기능	설명
동시성 제어	- 다수 트랜잭션의 동시 처리로부터 데이터 무결성 확보를 위한 제어 수행
회복 관리	- 시스템 오류 및 장애로 인한 데이터 손실 및 결함의 대응
성능 관리	- 데이터 처리 속도 확보를 위한 실행 계획의 최적화
보안 관리	- 비인가 사용자의 접근 제어 및 중요 정보의 암호화

### 4. 데이터베이스 종류

데이터베이스는 데이터를 관리하는 형태에 따라 계층형, 네트워크형, 관계형, 객체지향형, 객체관계형 데이터베이스가 존재한다.

〈표 3-3〉 데이터베이스 종류

종류	설명	특징
계층형 데이터베이스	<ul style="list-style-type: none"><li>- HDB: Hierarchical DB</li><li>- 데이터(Record)를 상하 종속적 관계로 계층화하여 관리하는 데이터 베이스</li></ul>	<ul style="list-style-type: none"><li>- 한 레코드는 필드로 구성되며 다른 레코드들의 포인터로 구성</li><li>- 빠른 접근 속도</li><li>- 데이터 변화에 대한 유연성 낮음</li></ul>
망형 데이터베이스	<ul style="list-style-type: none"><li>- NDB: Network DB</li><li>- 망형의 네트워크 구조로 데이터를 관리하는 데이터베이스</li></ul>	<ul style="list-style-type: none"><li>- 레코드에 부모 레코드의 포인터도 관리 가능</li><li>- 데이터 변화에 대한 유연성 좋음</li><li>- 데이터 모델링 복잡</li></ul>
관계형 데이터베이스	<ul style="list-style-type: none"><li>- RDB: Relational Database</li><li>- 데이터 간의 관계 구조를 관리하는 데이터베이스</li></ul>	<ul style="list-style-type: none"><li>- 레코드의 집합을 테이블로 구성</li><li>- 반복 그룹, 자료형 한계</li><li>- 동적 변화로 유연성 높음</li></ul>
객체지향형 데이터베이스	<ul style="list-style-type: none"><li>- OODB: Object Oriented DB</li><li>- 데이터를 객체화하여 관리하는 데이터베이스</li></ul>	<ul style="list-style-type: none"><li>- 객체 재사용, 캡슐화, 상속 가능</li><li>- 멀티미디어 지원 가능</li><li>- 데이터 모델링 복잡</li></ul>
객체관계형 데이터베이스	<ul style="list-style-type: none"><li>- ORDB: Object Relational DB</li><li>- 기존의 RDB에 객체의 개념을 적용한 데이터베이스</li></ul>	<ul style="list-style-type: none"><li>- 개발자가 데이터형 정의 가능</li><li>- 반복 그룹, 자료형 한계 극복</li><li>- 데이터 모델링 용이</li></ul>

## ② 트랜잭션(Transaction)

### 1. 트랜잭션의 개념

트랜잭션은 데이터베이스의 상태를 변화시키기 위한 최소 작업 단위로 한 번에 처리되어야 하는 질의어(SQL)의 묶음이다.

### 2. 트랜잭션의 특징

트랜잭션이 원활하게 처리되기 위해서는 <표 3-4>와 같이 ACID가 준수되어야 한다.

<표 3-4> 트랜잭션의 ACID 특징

특징	설명
Atomicity(원자성)	- 데이터베이스에 트랜잭션은 모두 반영되거나 전혀 반영되지 않아야 함
Consistency(일관성)	- 트랜잭션 시작 시점에 참조한 데이터는 종료까지 일관성을 유지해야 함
Isolation(고립성)	- 동시에 다수 트랜잭션이 처리되는 경우 서로의 연산에 개입하면 안 됨
Durability(지속성)	- 트랜잭션이 성공적으로 완료되면 처리 결과는 영속적으로 반영되어야 함

### 3. 트랜잭션의 고립화 수준(Transaction Isolation Level)

트랜잭션 동시에 처리되는 과정에서 트랜잭션 간에 Read, Write에 대해 허용하는 수준을 SQL 표준에서 4가지로 정의한다. 데이터베이스 관리시스템들은 이러한 수준을 제어할 수 있는 기능을 제공한다.

<표 3-5> 트랜잭션 고립화 수준

Isolation Level	설명
Read Uncommitted	- Commit되지 않은 데이터의 Read 허용
Read Committed	- 질의 시작 전 Commit된 데이터의 Read만 허용
Repeatable Read	- 트랜잭션 시작 전 Commit 된 데이터의 Read만 허용
Serializable	- 병행 처리되지 않고 순차적으로 처리되는 것과 동일한 수준

## 수행 내용 / 데이터베이스 특징 식별하기

---

### 재료 · 자료

- DBMS(Database Management System) 개발 가이드

### 기기(장비 · 공구)

- DBMS(Database Management System)
- 데이터베이스 개발 도구
- 자료 조사용 인터넷 검색 환경
- 문서 작성 도구, Presentation Tools, 뷔프로젝터

### 안전 · 유의 사항

- 다수의 학습자가 동시에 데이터베이스 테이블 조작 실습을 하는 경우 본인이 변경하지 않은 데이터가 조회되어 혼선이 있을 수 있다. 개별 실습을 하는 경우에는 테이블명을 학습자별로 중복되지 않도록 변경하여 실습한다.
- 자료를 조사하여 토론하는 학습 과정에는 적극적인 소수의 인원에게 의존하여 진행되지 않도록 모두 적극적으로 참여하고 균등하게 발표 기회를 갖도록 한다.

### 수행 순서

#### ① 데이터베이스 종류를 구분한다.

##### 1. 데이터베이스 개념을 학습한다.

데이터베이스는 데이터들의 집합체로 데이터를 시스템화하여 활용하기 위해 다양한 형태로 구성할 수 있다. 필요 지식 ①의 내용을 참고하여 데이터베이스의 개념을 학습한다.

##### 2. 데이터베이스의 종류를 조사하여 정리한다.

데이터를 관리하기 위해 초기에는 파일시스템(FileSystem)만 사용했지만 파일시스템은 다양한 데이터를 조합하여 분석하거나 변경 관리하는 것이 어려워 데이터베이스의 개념을 도입하게 되었다. 데이터베이스는 계층형 데이터베이스 형태로 시작되어 재사용이 가능하고 유연한 구조를 갖는 객체관계형 데이터베이스로 발전하였다. 필요 지식 ②의 내용을 참고하여 데이터베이스의 종류를 조사하고 각각의 특징을 파악하여 정리한다.

##### 3. 데이터베이스의 종류를 구분한다.

학습 2-2의 수행 내용 ①에서 설치한 PostgreSQL의 다양한 특징을 조사하고, 위에서 학습한 데이터베이스 종류와 비교하여 PostgreSQL의 데이터베이스 종류를 구분한다.

(1) PostgreSQL의 주요 특징을 조사하여 정리한다.

〈표 3-6〉 PostgreSQL 특징 정리 예시

특징	설명
표준 SQL	- 대부분 ANSI SQL:2008 표준 기반으로 한 질의어 사용
ACID 지원	- 트랜잭션 수행 시 보장되어야 하는 ACID(Atomicity, Consistency, Isolation, Durability) 특성 지원
테이블 상속 활용	- 테이블 생성 시 다수 테이블을 상속받아 생성 가능하며, 상위 테이블 조회 시 상속받은 하위 테이블 정보 포함 조회 가능
확장 모듈 활용	- PostGIS(Geographic Information System), HStore(Key-Value), 암호화 등 확장하여 활용 가능
객체 활용	- 사용자가 연산자, 복합 자료형 등을 DB 객체로 정의하여 활용 가능
PL/pgSQL	- 프로그래밍 언어를 이용해 함수 생성 가능

수행 tip

- 대부분의 소프트웨어는 개발자들이 활용할 수 있도록 가이드 문서를 제공한다. PostgreSQL의 개발 가이드를 참고하여 주요 기능 및 활용법을 학습할 수 있다.  
(<https://postgresql.kr/docs/9.4/index.html>)

(2) 데이터베이스 종류별 특징과 PostgreSQL의 특징을 비교한다.

각자 조사하여 정리한 데이터베이스 종류와 PostgreSQL의 특징을 모둠별로 공유하고 토의를 통해 매핑(Mapping)한다.

(3) 데이터베이스의 종류를 구분한다.

PostgreSQL은 SQL을 기반으로 하며 트랜잭션의 특성인 ACID를 지원하는 관계형 데이터베이스의 특징과 테이블 상속 등 객체지향 데이터베이스의 특징을 모두 가지고 있는 객체관계형 데이터베이스라고 구분할 수 있다.

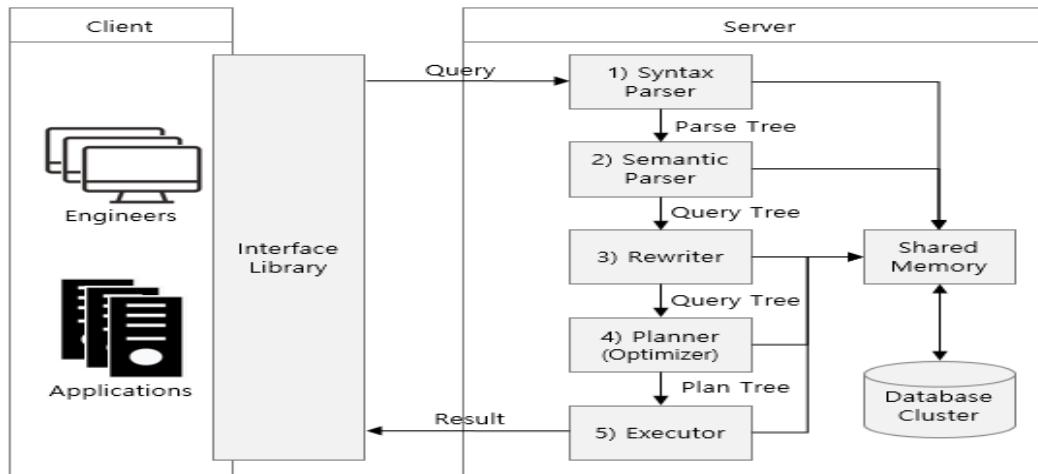
② 응용소프트웨어 개발에 필요한 데이터베이스 특징을 식별한다.

필요 지식 ①-2의 내용과 같이 데이터베이스는 실시간 접근성(Real-time Accessibility), 계속적인 변화(Continuous Evolution), 동시 공용(Concurrent Sharing), 내용에 의한 참조(Contents Reference)의 특성을 가지고 있다. 이런 특성을 고려하여 응용소프트웨어 개

발을 지원하기 위해 PostgreSQL이 가지고 있는 특징을 식별한다.

### 1. 실시간 접근성을 제공하기 위한 특징을 식별한다.

데이터베이스 관리시스템은 빠른 속도로 질의어를 처리하기 위해 다양한 메커니즘을 이용하여 이를 통해 실시간 접근성을 제공한다. 아래 [그림 3-2]를 참고하여 PostgreSQL 질의어 처리 과정을 분석하고, 빠른 성능 확보를 위한 메커니즘을 조사하여 실시간 접근성 제공을 위한 특징을 식별한다.



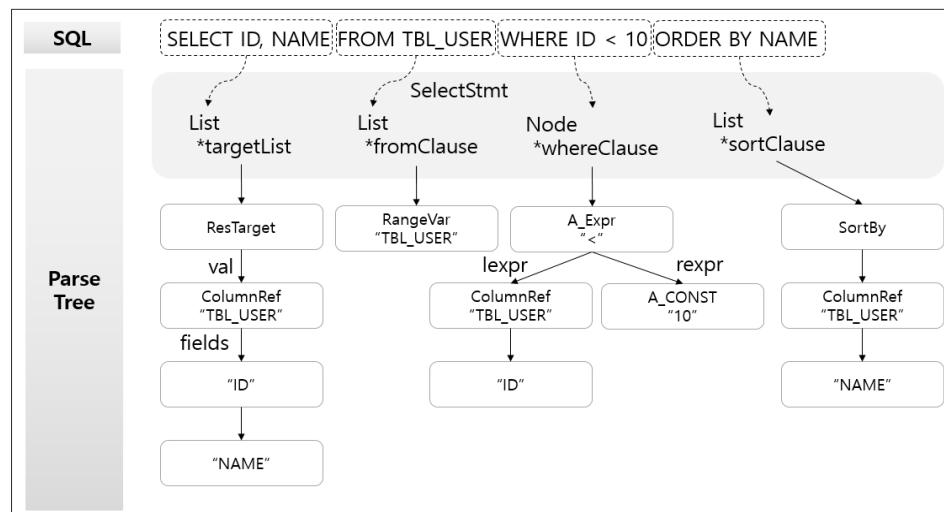
출처: 집필진 제작(2021)

[그림 3-2] PostgreSQL 질의어 처리 과정

### (1) PostgreSQL 질의어 처리 과정을 분석한다.

#### (가) Syntax Parser 처리 과정을 분석한다.

Select 구문에 문법적 오류가 있는지 체크하고 정상이면 [그림 3-3]과 같이 Select, From, Where, 정렬 구문으로 분할하여 Parse Tree를 생성한다.

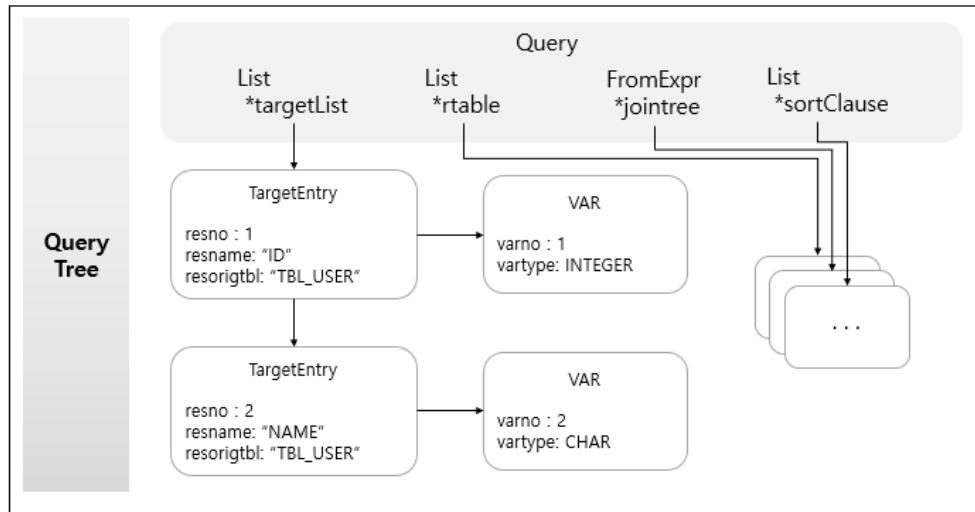


출처: 집필진 제작(2021)

[그림 3-3] Parse Tree 생성 예시

(나) Semantic Parser 처리 과정을 분석한다.

Parse Tree의 내용을 분석하여 실제 존재하는 테이블, 컬럼인지 확인하고 정상이면 [그림 3-4]와 같이 각 Entry의 속성을 나타내는 metadata를 추가하여 Query Tree를 생성한다.



출처: 집필진 제작(2021).  
[그림 3-4] Query Tree 생성 예시

(다) Planner 처리 과정을 분석한다.

최적화된 실행 계획을 생성하기 위해 pg\_rules 시스템을 이용해 사전에 정의된 규칙(Rules) 기반으로 Query Tree를 재조정한다.

**수행 tip**

- 위에서 예제로 든 간단한 쿼리로는 변화를 표현하기 어렵고 복잡한 쿼리를 예제로 이용하면 이해가 어렵기 때문에 본 과정에서는 제외한다. 차후에 View, Join을 이용하는 경우를 예시로 심화 학습하도록 한다.

(라) Rewriter 처리 과정을 분석한다.

질의어를 실행하는 경로(Path)는 다양하다. 테이블을 앞에서부터 읽을 수도 있고 뒤에서부터 읽을 수도 있으며, 인덱스를 먼저 읽어서 테이블의 원하는 데이터로 바로 이동할 수도 있다. 다양한 경로 중에 최소 비용(Cost)이 소요되는 경로를 선택하여 Plan Tree를 생성한다.

### 수행 tip

- 생성된 Plan Tree는 EXPLAIN command로 확인할 수 있다. 응용소프트웨어 개발을 위해서는 Plan Tree를 읽고 최적의 Plan Tree가 선정되도록 질의어를 만들 수 있어야 한다. 차후에 심화 학습하도록 한다.

#### (마) Executer 처리 과정을 분석한다.

Plan Tree를 기반으로 메모리에 로딩된 데이터를 이용해 쿼리를 수행하고, 메모리에 로딩되지 않는 경우 디스크상의 데이터를 메모리에 로딩하여 처리하고 그 결과를 Client에 반환한다.

#### (2) 실시간성을 확보하기 위한 논리적/물리적 메커니즘을 분석한다.

데이터베이스 관리시스템이 실시간성을 확보하기 위한 PostgreSQL의 논리적/물리적 메커니즘을 조사하여 정리한다.

##### (가) 논리적인 메커니즘을 분석하여 정리한다.

논리적인 메커니즘을 고려하여 SQL 작성할 수 있도록 분석하여 정리한다.

〈표 3-7〉 실시간성 확보를 위한 PostgreSQL의 논리적 메커니즘 정리 예시

논리적 메커니즘	설명
Soft Parsing	- Syntax Parser가 생성한 Parse Tree가 기존에 생성한 것과 동일한 경우에는 Query Tree, Plan Tree 생성 과정을 생략하고 이미 생성되어 있는 Plan Tree를 이용해 바로 실행하여 시간을 단축시킴
Index	- 자주 사용되는 데이터의 위치 정보를 인덱스로 별도 관리하여 데이터 검색 속도를 단축시킴
Hint	- PG_HINT_PLAN 기능으로 Plan Tree 변경

##### (나) 물리적 메커니즘을 분석하여 정리한다.

물리적 메커니즘을 제어할 수 있는 설정값을 조정할 수 있도록 분석하여 정리한다.

〈표 3-8〉 실시간성 확보를 위한 PostgreSQL의 물리적 메커니즘 정리 예시

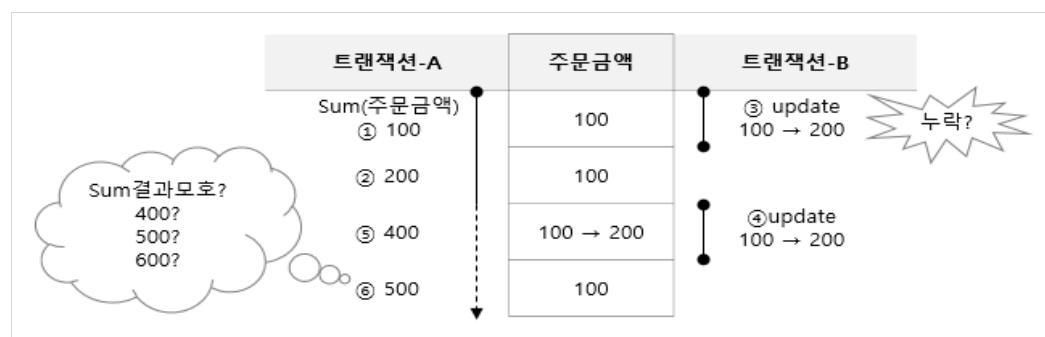
물리적 메커니즘	설명
Shared Buffers	- Disk에 저장된 데이터를 탐색하는 속도가 느리기 때문에 SQL 실행 결과를 Shared Memory의 Shared Buffers에 저장하여 재사용
Work Memory	- 정렬, 해시/머지 조인 처리 과정에서 데이터를 임시로 저장하는 경우 Disk I/O 방지를 위해 Local Memory의 Work Memory 사용
Temp Buffers	- User Session에서 사용하는 임시 테이블 관리용 Local Memory

## 2. 계속적으로 변화하는 데이터베이스를 관리하기 위한 특징을 식별한다.

데이터베이스의 데이터는 삽입(Insert), 삭제(Delete), 갱신(Update) 처리에 의해 계속 변화하기 때문에 데이터베이스 관리시스템은 이로 인해 데이터 처리에 모순이 발생되지 않도록 관리한다. 계속적으로 변화하는 특성으로 인해 발생 가능한 문제점에 대해 토론해보고 이를 방지하기 위한 PostgreSQL의 특징을 식별한다.

### (1) 계속적으로 변화하는 데이터베이스의 특성으로 인한 문제점을 토론한다.

[그림 3-5]과 같이 트랜잭션-A가 숫자 컬럼의 값을 합산하고 있는 과정에서 트랜잭션-B가 특정 값을 변경하는 경우 트랜잭션-A의 합산 결과에는 모순이 있을 수 있다. 이와 같이 발생 가능한 문제들에 대해 토론한다.

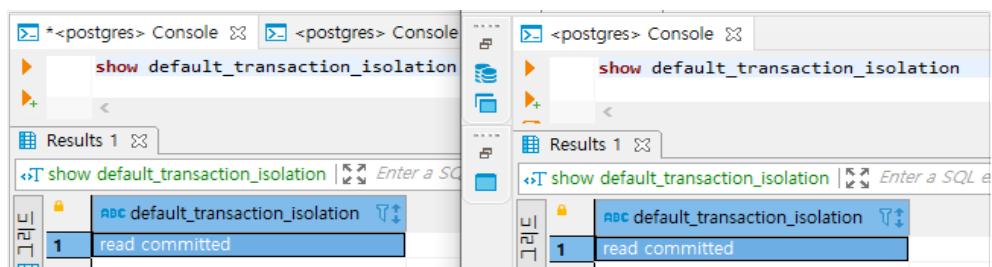


출처: 집필진 제작(2021)  
[그림 3-5] 계속적 변화로 인한 모호성 예시

### (2) 계속적으로 변화하는 데이터베이스의 특성을 관리하기 위한 PostgreSQL의 특징을 조사하여 정리한다.

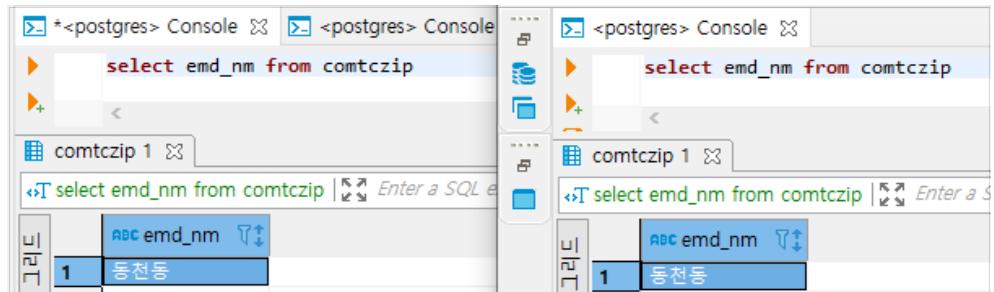
계속적으로 변화하는 데이터베이스의 특성을 관리하기 위해 다양한 기법들 중 가장 기본이 되는 트랜잭션에 대해 필요 지식 ②를 참고하여 학습한다. 특히, 고립화 수준(Transaction Isolation Level)을 제어함으로써 트랜잭션 처리 과정에서 참조되는 의 일관성을 유지하여 트랜잭션 결과의 모호성을 방지할 수 있다. 고립화 수준을 확인하고 고립화 수준에 따른 트랜잭션의 차이를 실습한다.

#### (가) show transaction isolation level 명령어로 Isolation Level을 확인한다.



출처: 집필진 제작(2021)  
[그림 3-6] 고립화 수준 실습 - 고립화 수준 확인

(나) 동일한 값을 조회한다.



```
select emd_nm from comtczip
```

emd_nm
동천동

```
select emd_nm from comtczip
```

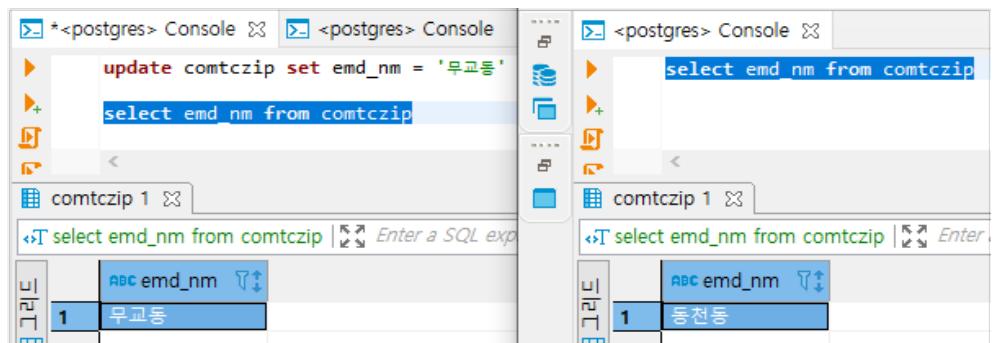
emd_nm
동천동

출처: 집필진 제작(2021)

[그림 3-7] 고립화 수준 실습 - 변경 전 확인

(다) 데이터 변경 후 동일한 값을 재조회한다.

오른쪽 트랜잭션은 변경 전 값이 그대로 조회된 것을 확인할 수 있다.



```
update comtczip set emd_nm = '무교동'
```

```
select emd_nm from comtczip
```

emd_nm
무교동

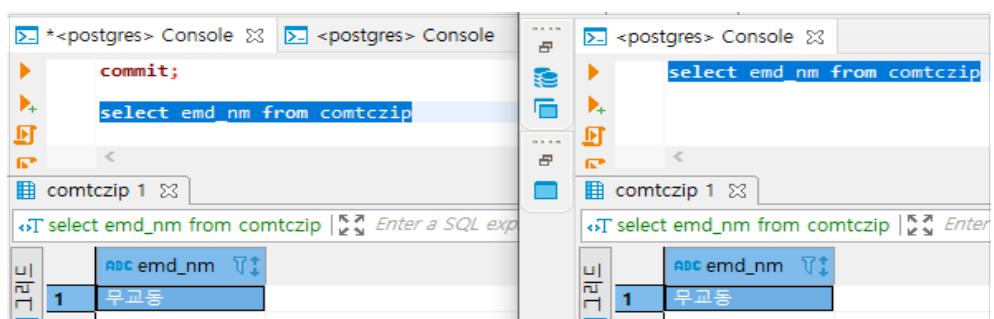
```
select emd_nm from comtczip
```

emd_nm
동천동

출처: 집필진 제작(2021)

[그림 3-8] 고립화 수준 실습 - 변경 후 확인

(라) commit 후 동일한 값을 재조회한다.



```
commit;
```

```
select emd_nm from comtczip
```

emd_nm
무교동

```
select emd_nm from comtczip
```

emd_nm
무교동

출처: 집필진 제작(2021)

[그림 3-9] 고립화 수준 실습 - commit 후 확인

### 수행 tip

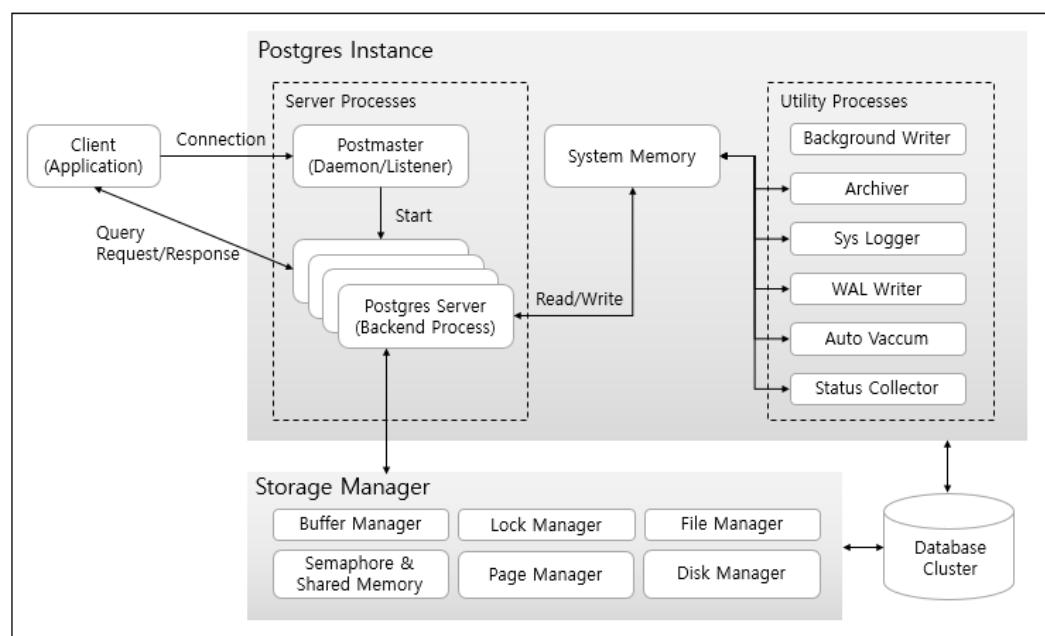
- PostgreSQL은 Read committed, Repeatable read, Serializable 3가지 고립화 수준을 제공한다. Set Transaction 명령어를 이용해 고립화 수준을 변경 해 심화 실습한다.

### 3. 동시 공용으로 사용하는 데이터베이스를 관리하기 위한 특징을 식별한다.

데이터베이스 관리시스템은 동시에 접속하는 다수의 사용자 및 응용소프트웨어가 데이터베이스를 원활하게 사용할 수 있도록 Multi-Process를 지원하고 Process가 자원을 공유하거나 분배하여 사용할 수 있는 시스템 아키텍처를 제공한다. PostgreSQL이 제공하는 프로세스 아키텍처를 분석하여 특징을 식별하여 정리한다.

#### (1) PostgreSQL의 프로세스 아키텍처를 분석한다.

아래 [그림 3-10]을 참고하여 PostgreSQL의 프로세스 아키텍처를 분석한다. PostgreSQL 기동 시 server process가 가장 먼저 시작되어 client의 요청에 따라 backend process를 기동하고 connection을 생성한다. backend process들은 background writer와 같은 background process들을 관리하고 Local memory를 할당받아 각 client의 요청을 처리한다. 모든 프로세스가 공유해서 사용하는 shared memory는 server process가 관리한다.



출처: 집필진 제작(2021)  
[그림 3-10] PostgreSQL의 프로세스 아키텍처

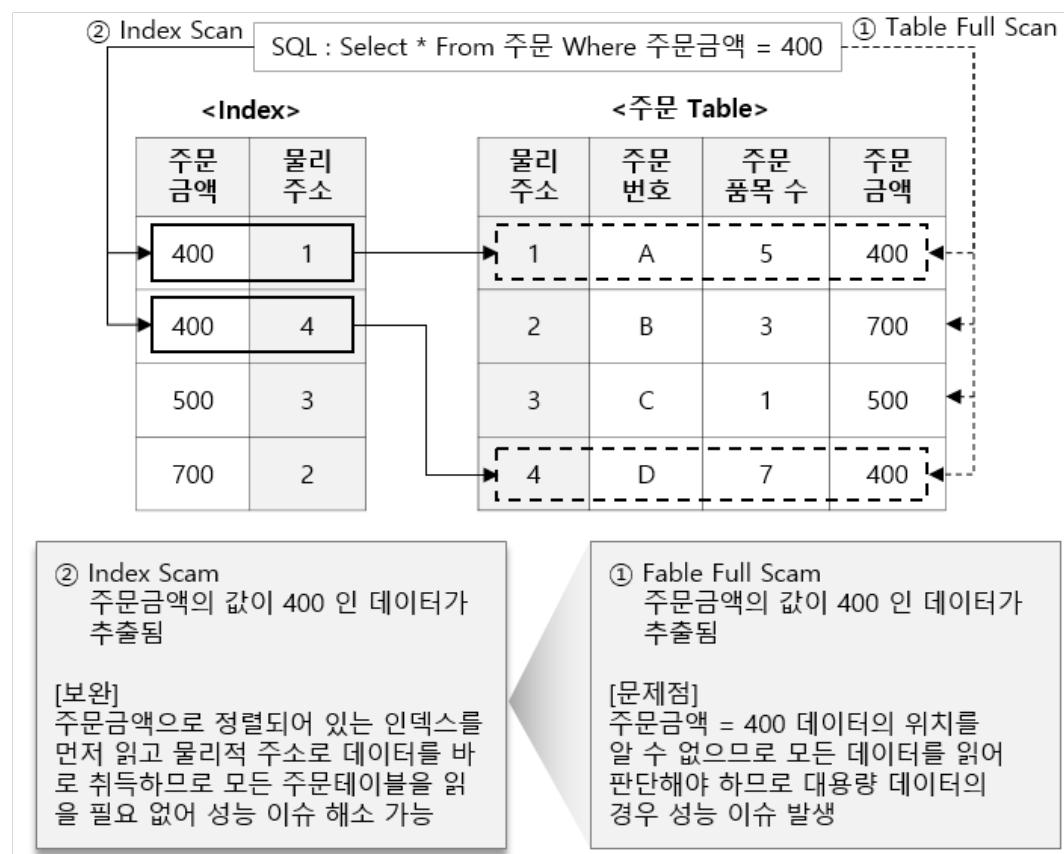
#### (2) 동시 공용으로 인한 데이터베이스 특징을 식별하여 정리한다.

모듈별로 분석한 내용을 공유하고 토론을 통해 Multi-Process 환경에 대한 특징을 정리한다.

### 4. 데이터 참조 시 데이터의 주소가 아닌 내용을 기준으로 하는 데이터베이스 특성을 식별한다.

데이터베이스의 질의어는 데이터의 주소가 아닌 저장되어 있는 데이터의 내용에 의

해 처리된다. 이러한 특성으로 인해 아래 [그림 3-11]의 ①과 같이 모든 데이터를 읽어야 하므로 성능 저하의 문제가 발생될 수 있으므로 ②와 같이 인덱스를 먼저 읽어 데이터의 범위를 축소하는 등의 메커니즘이 반영되어 있다. 이러한 특징들을 모둠별로 조사하고 내용을 공유하여 정리한다.



출처: 집필진 제작(2021)  
[그림 3-11] 내용에 의한 참조 예시

## 3-2. 관계형 데이터베이스 테이블 정의

### 학습 목표

- 주어진 E-R(Entity-Relationship) 다이어그램을 이용하여 관계형 데이터베이스의 테이블을 정의할 수 있다.

### 필요 지식 /

#### ① 데이터 모델링

##### 1. 데이터 모델링 개념

시스템으로 구성하기 위한 데이터의 집합(실체)을 도출한 후 각 집합을 구성하는 세부 속성과 식별자를 정의하고 각 데이터 집합 간의 관계를 정해진 표기법(Notation)으로 시각화하는 과정이다.

##### 2. 데이터 모델링 프로세스

데이터 모델링은 요구사항 수집 및 분석을 통해 도출된 데이터 집합을 이용해 개념, 논리, 물리 모델링 과정을 통해 데이터베이스를 구현한다.

〈표 3-9〉 데이터 모델링 유형

프로세스	설명
1. 요구사항 수집/분석 개념 모델링	- 사용자 및 데이터베이스 용도를 식별하고 사용자 요구사항을 수집 및 분석하여 요구사항 정의서를 작성하는 과정
	- 현실세계의 정보를 추상화하여 주제 영역을 정의하고 식별자/관계/속성을 도출해 개념 ERD(Entity-Relationship Diagram) 작성
2. 설계 논리 모델링 물리 모델링	- 개념 데이터 모델을 특정 데이터베이스(계층형, 망형, 관계형)에 적합하도록 구조화하여 논리 ERD를 작성하는 과정
	- 특정 DBMS(Database Management System)에서 활용 가능하도록 물리 ERD 및 테이블 정의서 작성 과정
3. 데이터베이스 구현	- 물리 ERD 및 테이블 정의서를 이용해 특정 DBMS에 데이터베이스를 구축하는 과정

#### ② E-R(Entity-Relationship) 데이터 모델

##### 1. E-R 데이터 모델의 개념

현실세계의 구성요소들을 데이터베이스로 관리하기 위해 유형화(Classification), 집단화(Aggregation), 일반화(Generalization) 과정을 통해 추상화(ABstraction)하여 개체(Entity)와 관계(Relationship)로 구조화한 데이터 모델이다.

## 추상화(Abstraction)

현실세계에 존재하는 개체(Entity)들의 특징을 유형화(Classification), 집단화(Aggregation), 일반화(Generalization)하여 추상적 개념으로 표현하는 과정

- 유형화(Classification): 동일한 특성을 하나의 유형(Class)로 분류하여 속성으로 정의  
[예시] 인사담당자, 영업담당자, 구매담당자를 담당자구분으로 유형화
- 집단화(Aggregation): 연관된 속성을 하나의 집단으로 분류하여 새로운 속성으로 정의  
[예시] 담당자구분, 부서, 이름, 전화번호를 담당자로 집단화
- 일반화(Generalization): 공통 속성을 파악하여 전체집합과 부분집합으로 분류  
[예시] 이름, 전화번호를 사람으로 일반화

## 2. E-R(Entity-Relationship) 데이터 모델의 구성요소

〈표 3-10〉 E-R(Entity-Relationship) 데이터 모델 구성요소

구성요소	설명
개체(Entity)	<ul style="list-style-type: none"><li>- 사람, 사물, 사건, 개념 등의 유무형의 특성을 공유하는 독립적인 실체로 인스턴스(Instance, 값)의 집합</li><li>- 유일한 식별자에 의해 식별 가능해야 하며 반드시 하나 이상의 속성(Attribute), 하나 이상의 관계(Relation)로 구성되어야 함</li></ul>
속성(Attribute)	<ul style="list-style-type: none"><li>- 개체(Entity)를 구성하는 특성</li></ul>
관계(Relationship)	<ul style="list-style-type: none"><li>- 개체 간의 상호 연관성을 표현하는 페어링(Paring)의 집합</li></ul>

## 3. E-R(Entity-Relationship) 데이터 모델 표기법

### (1) E-R 데이터 모델 표기법 유형

〈표 3-11〉 E-R(Entity-Relationship) 데이터 모델 표기법 유형

유형	설명
Chen 표기법	<ul style="list-style-type: none"><li>- 1976년 Peter Chen에 의해 만들어진 최초의 표기법</li><li>- 교육용으로 주로 사용되며 실무적으로는 사용하지 않음</li></ul>
정보공학(IE) 표기법	<ul style="list-style-type: none"><li>- 1981년 Clive Finkelstein과 James Martin에 의해 발표되고 1980년 중반 James Martin에 의해 체계가 정리된 표기법</li></ul>
바커(Baker) 표기법	<ul style="list-style-type: none"><li>- 1986년 Richard Barker, Ian Palmer, Harry Ellis 등에 의해 개발되고 이후 Richard Barker에 의해 자속적으로 개선된 표기법</li></ul>

### (2) E-R 데이터 모델 표기법

실무에서 주로 사용되는 정보공학(IE) 표기법과 바커(Barker) 표기법을 알아본다.

#### (가) 개체(Entity) 표기법

개체는 다른 개체에 의존하지 않고 독립적으로 존재 가능한 독립 개체와 독립 개체에 종속되는 종속 개체가 존재한다.

〈표 3-12〉 개체(Entity) 표기법

구분	정보공학(IE)	바커(Barker)						
독립 개체	<p>두 개의 칸으로 나눠진 모서리가 직각 인 사각형으로 표현</p> <p style="text-align: center;"><b>Entity_Name</b></p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>Primary Key</td></tr> <tr><td>Attribute_Name_1</td></tr> <tr><td>Attribute_Name_2</td></tr> </table>	Primary Key	Attribute_Name_1	Attribute_Name_2	<p>모서리가 둥근 소프트 박스(Soft Box) 로 표현</p> <p style="text-align: center;"><b>Entity_Name</b></p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>Primary Key</td></tr> <tr><td>Attribute_Name_1</td></tr> <tr><td>Attribute_Name_2</td></tr> </table>	Primary Key	Attribute_Name_1	Attribute_Name_2
Primary Key								
Attribute_Name_1								
Attribute_Name_2								
Primary Key								
Attribute_Name_1								
Attribute_Name_2								
	가. 정보공학 표기법 - 독립 개체							
종속 개체	<p>두 개의 칸으로 나눠진 모서리가 둥근 사각형으로 표현</p> <p style="text-align: center;"><b>Entity_Name</b></p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>Primary Key</td></tr> <tr><td>Attribute_Name_1</td></tr> <tr><td>Attribute_Name_2</td></tr> </table>	Primary Key	Attribute_Name_1	Attribute_Name_2	<p>다. 바커 표기법 - 독립/종속 개체</p> <p style="text-align: center;"><b>Entity_Name</b></p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>Primary Key</td></tr> <tr><td>Attribute_Name_1</td></tr> <tr><td>Attribute_Name_2</td></tr> </table>	Primary Key	Attribute_Name_1	Attribute_Name_2
Primary Key								
Attribute_Name_1								
Attribute_Name_2								
Primary Key								
Attribute_Name_1								
Attribute_Name_2								
	나. 정보공학 표기법 - 종속 개체							

(나) 속성(Attribute) 표기법

속성은 반드시 값이 존재해야 하는 필수(Mandatory) 속성과 선택(Optional) 속성이 존재하며, 개체에서 하나의 인스턴스를 식별할 수 있는 하나 이상의 속성을 유일식별자(Unique Identifier) 또는 주키(PK, Primary key)라고 한다. 또한, 다른 개체에서 상속받은 하나 이상의 속성은 외래식별자(Foreign Identifier) 또는 외래키(FK, Foreign Key)라고 한다.

〈표 3-13〉 속성(Attribute) 표기법

구분	정보공학(IE)	바커(Barker)
필수	구별하지 않음	속성 앞에 * 기호 표기
선택	개체 상단 박스에 표기	속성 앞에 o 기호 표기
PK	속성 뒤에 (PK) 표기	속성 앞에 # 기호 표기
FK		속성 뒤에 (FK) 표기

예시

**Entity\_Name**

Primary Key
Attribute_Name_1
Attribute_Name_2
Attribute_Name_3 (FK)

가. 정보공학 표기법 - 속성

**Entity\_Name**

# Primary Key
* Attribute_Name_1
o Attribute_Name_2
o Attribute_Name_3 (FK)

나. 바커 표기법 - 속성

#### (다) 관계(Relationship) 표기법

관계는 엔티티 간의 상호 연관 규칙을 표현하는 방법으로 하나의 인스턴스가 몇 개의 인스턴스와 연관이 되는지를 관계 기수성(Cardinality, Degree)으로 표현하고, 관계의 필수/선택 여부를 관계 선택성으로 표현하며, 관계로 인스턴스를 식별할 수 있는지 여부를 관계 식별성으로 표현한다.

〈표 3-14〉 관계(Relationship) 표기법

구분	정보공학(IE)	바커(Barker)
기수성	1 라인 (-) 표기	라인 (-) 표기
	N 까마귀발(Crow's Foot) 표기	까마귀발(Crow's Foot) 표기
선택성	필수 동그라미 생략	실선 표기
	선택 동그라미 표기	점선 표기
식별성	식별 실선 표기	UID Bar ( ) 표기
	비식별 점선 표기	UID Bar ( ) 생략
예시	1:1	-----+-----+
	필수	-----
	비식별	가. 정보공학 표기법 - 관계(1)
	N:N	다. 바커 표기법 - 관계(1)
선택	선택	---
	식별	나. 정보공학 표기법 - 관계(2)
식별	식별	라. 바커 표기법 - 관계(2)

### ③ 관계형 데이터베이스 테이블

#### 1. 관계형 데이터베이스 테이블 개념

데이터 모델링을 통해 도출된 개체(Entity)와 관계(Relationship)를 데이터베이스에서 관리하기 위한 2차원 표 형태의 저장 공간

#### 2. 관계형 데이터베이스 테이블 구성

〈표 3-15〉 속성(Attribute) 표기법

구성	설명
필드	열과 행이 교차하는 지점 하나의 값
열	개체를 구성하는 속성(Attribute)
행	하나의 인스턴스(Instance)를 구성하는 속성 값의 집합으로 튜플(Tuple), 레코드(Record)라고도 함

## 수행 내용 / 관계형 데이터베이스 테이블 정의하기

### 재료·자료

- 관계형 데이터베이스 테이블 정의 실습 시나리오

### 기기(장비 · 공구)

- E-R(Entity-Relationship) 다이어그램 도구
- 문서 작성 도구, Presentation Tools, 빔 프로젝터

### 안전 · 유의 사항

- E-R 다이어그램 솔루션을 이용하는 경우 솔루션에 따라 표준화된 E-R 데이터 모델 표기법과 일부 다른 부분이 있을 수도 있다. 솔루션 특징으로 이해하고 실습하도록 한다.

### 수행 순서

#### ① 사용자의 요구사항을 분석하여 데이터 모델링을 수행한다.

사용자의 요구사항을 분석하여 데이터 모델링을 수행하고 E-R 다이어그램을 작성한다. 데이터 모델링은 필요 지식 ①-2에서 학습한 것과 같이 개념/논리/물리 단계로 구분되어 있지만 시스템의 규모에 따라 동시에 수행할 수도 있다. 이번 학습은 실습이 가능하도록 간단한 상황을 예시로 수행하기 때문에 단계 구분을 최소화하고 순차적으로 수행한다.

##### 1. 사용자의 요구사항을 분석한다.

###### (1) 아래 데이터베이스 구축 요청 사례를 읽고 요구사항을 파악한다.

###### 데이터베이스 구축 요청 사례

트럭 렌트 회사에서 렌탈 업무를 관리하기 위한 시스템을 구축하고자 한다. 본 회사는 300개의 렌탈 사무소를 가지고 있으며 5가지 종류의 차량을 5,000대 구비하고 있다. 렌탈 사무소는 숫자 3자리의 고유한 사무소 번호로 구별되며 사무소, 주소, 전화번호를 공유하여 사무소 간에 연락을 할 수 있어야 한다. 차량은 1t, 5t, 10t, 24t, 36t 트럭을 가지고 있으며 차량 관리를 위하여 마지막 정비 일자, 등록 말소일을 관리해야 한다. 렌탈 계약은 한 번에 1대만 가능하며 개인 고객, 법인 고객과 맺을 수 있다. 렌탈 계약 시 고유한 계약 번호가 자동으로 발급되며 렌탈한 사무소와 트럭에 기록된 주행거리를 기록하고 반납 시에 반납한 사무소와 변경된 주행거리를 기록한다. 법인은 사업자 번호로 식별할 수 있으며 법인명, 주소, 연락처를 관리한다. 개인 고객은 휴대폰 번호로 식별할 수 있으며 고객명, 주소를 관리한다.

###### (2) 요구사항을 분석하여 요구사항 정의서를 작성한다.

트럭 렌트 회사의 렌탈 업무 관리 시스템 구축을 위한 요구사항 정의서를 <표

3-16) 예시를 참고하여 작성한다.

〈표 3-16〉 요구사항 정의서 작성 예시

순번	구분	요청내역	요청일	요청자	상태
1	사무소 관리	숫자 3자리 고유 사무소 번호, 주소, 전화 번호 관리 필요	7/17	홍길동	접수
2	차량 관리	트럭 종류(1t, 5t, 10t, 24t, 36t), 마지막 정비 일자, 등록 말소일 관리 필요	7/17	홍길동	접수
3	고객 관리	개인 고객은 휴대폰 번호로 식별, 고객명, 주소 관리 필요	7/17	홍길동	접수
4	고객 관리	법인 고객은 사업자 등록 번호로 식별, 법인명, 주소, 연락처 관리 필요	7/17	홍길동	접수
5	렌탈 계약 관리	렌탈 계약은 차량 1대만 가능하며 렌탈 계약 시 고유 계약 번호 자동 발급 필요	7/17	홍길동	접수
6	렌탈 계약 관리	렌탈 계약 시 렌탈 사무소와 차량의 주행 거리 기록 필요	7/17	홍길동	접수
7	렌탈 계약 관리	반납 시 반납 사무소와 차량의 변경된 주행 거리 기록 필요	7/17	홍길동	접수

2. 개념/논리 데이터 모델링을 수행한다.

(1) 주제 영역을 정의한다.

요구사항 분석 결과를 토대로 주제 영역을 선정한다.

〈표 3-17〉 주제 영역 정의서 작성 예시

분류	주제 영역명	설명	대표 개체
렌탈	렌탈 계약	개인 또는 법인 고객에게 트럭 대여를 위한 계약 관리	사무소, 차량, 고객, 계약

### 주제 영역

데이터를 구조화하여 표현하기 위한 최상위 집합으로 하위 수준으로 분할하여 하위 주제 영역 또는 개체(Entity)로 계층화하여 표현한다.

(2) 개체(Entity)를 정의한다.

렌탈 계약 업무를 구성하는 핵심 개체를 선정하고 식별자(Identifier)를 구분하여 개체정의서를 작성한다.

〈표 3-18〉 개체(Entity) 정의서 작성 예시

개체명	설명	분류	식별자	관련 속성
사무소	회사에서 관리하는 사무소 정보	기본	사무소 번호	주소, 전화번호
차량	회사에서 관리하는 차량 정보	기본	차량 번호	차량 종류, 마지막 정비 일자, 등록 말소일
고객	계약을 체결하는 고객 정보	기본	고객 번호	고객명, 주소, 휴대폰 번호, 고객 구분
계약	차량의 렌탈 계약 정보	중심	계약 번호	차량 번호, 시작 사무소 번호, 시작 주행거리, 종료 사무소 번호, 종료 주행거리

### 개체(Entity) 분류

개체는 발생 시점에 따라 기본, 중심, 행위 개체로 분류한다.

- 기본 개체: 독립적으로 생성되어 타 개체의 부모 역할을 수행
- 중심 개체: 업무에서 중심적인 역할을 수행하는 개체로 기본 개체에서 발생
- 행위 개체: 개체의 활동에 의해 생성된 원천(Raw) 정보로 발생순서 중요

### 수행 tip

- 개체는 업무 담당자가 실무에 사용하는 문서를 참조하거나 담당자 인터뷰, 업무 프로세스 분석 등을 통해 실무에 사용하는 용어로 도출한다.
- 개체 도출 후에는 순수성, 동질성, 독립성 등의 측면에서 적합성을 검증해야 한다.

### (3) 속성(Attribute)을 정의한다.

속성을 도출하고 각 속성에 대해 속성명, 도메인, 필수 여부를 지정한다. 도메인은 데이터 타입, 길이, 허용값, 디폴트 값 등으로 구성된다.

### 수행 tip

- 속성은 최소 단위까지 분할해야 하며 하나의 값만 갖도록 구성해야 한다.
- 최소 단위까지 분할 이후에 관리 편의성을 고려하여 우편번호, 일자 등을 통합하는 것을 검토한다.

### (4) 관계(Relationship)를 정의한다.

도출된 개체 간의 관계를 기수성, 선택성, 식별성 측면에서 분석하여 정리한다. 관계를 분석할 때는 상호간의 연관성을 파악해야 하므로 양방향에서 파악한다.

〈표 3-19〉 관계(Relationship) 정리 예시

개체	구분	관계	설명
기수성	1:N		사무소는 다수의 계약을 체결한다. 계약은 하나의 사무소를 지정한다.
사무소 - 계약	선택성	필수:선택	사무소는 계약을 안 맺을 수 있음 계약은 반드시 사무소를 지정해야 함
식별성	비식별:비식별		사무소의 식별자로 계약을 식별할 수 없음 계약의 식별자로 사무소를 식별할 수 있음

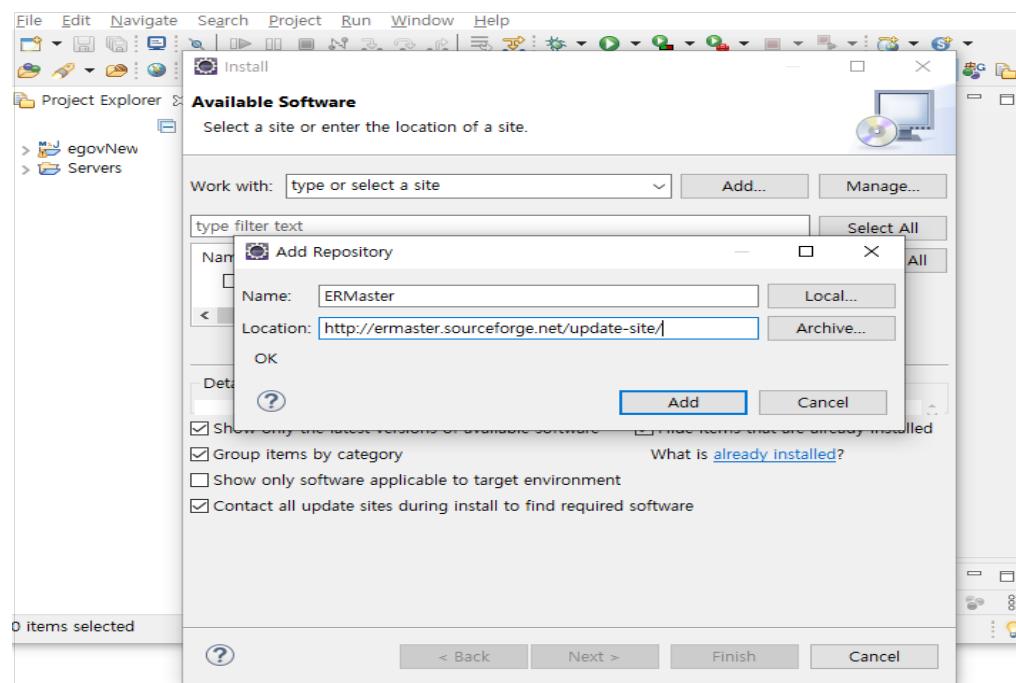
### 3. E-R(Entity-Relationship) 다이어그램을 작성한다.

E-R 다이어그램은 필요 지식 2-3에서 학습한 것과 같이 다양한 표기법이 존재하며 표기법에 따라 활용 가능한 도구가 상이하다. 이번 학습에는 Eclipse에 확장하여 활용 가능한 Apache 2.0 라이선스 오픈소스 도구인 ERMaster를 활용하여 정보공학(IE) 표기법으로 다이어그램을 작성한다.

#### (1) E-R 다이어그램 작성 도구를 설치한다.

이클립스를 실행하여 화면 상단 메뉴의 Help > Install New Software > Add 버튼을 클릭하고 Name과 Location을 입력하고 Next 버튼과 Finish 버튼을 클릭하여 설치한다. 정상적으로 설치되면 이클립스가 재기동된다.

Location: <http://ermaster.sourceforge.net/update-site/>

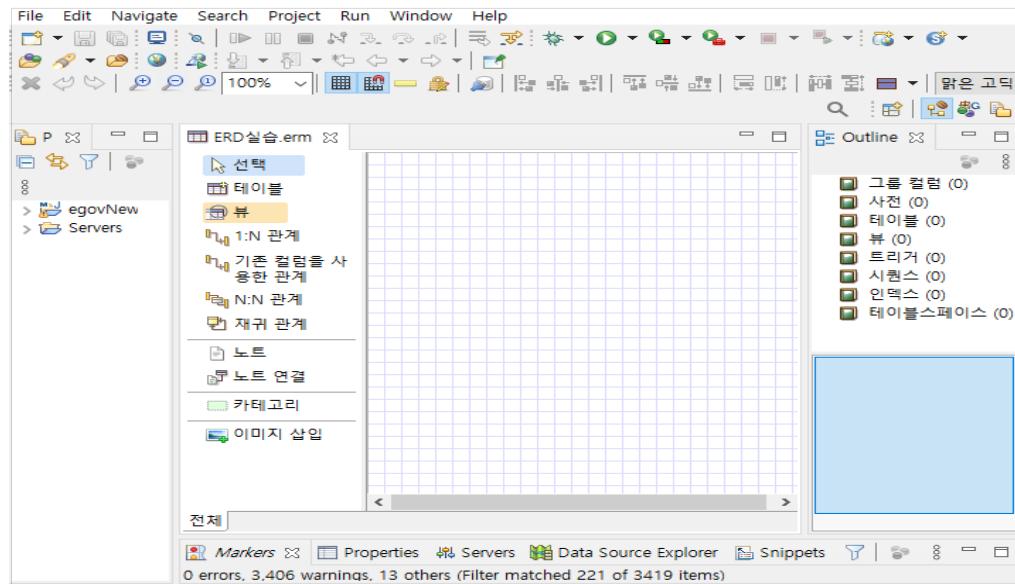


출처: 집필진 제작(2021)

[그림 3-12] E-R(Entity-Relationship) 다이어그램 작성 도구 설치

### (2) E-R 다이어그램 작성 도구를 실행한다.

이 클립스를 실행하여 화면 상단의 File > New > Other 메뉴에서 ERMaster를 선택하고 Next를 클릭한다. 새 ER 다이어그램의 Folder와 File Name을 입력한 후 Next를 클릭하여 PostgreSQL을 선택하고 Finish를 클릭하여 실행한다.

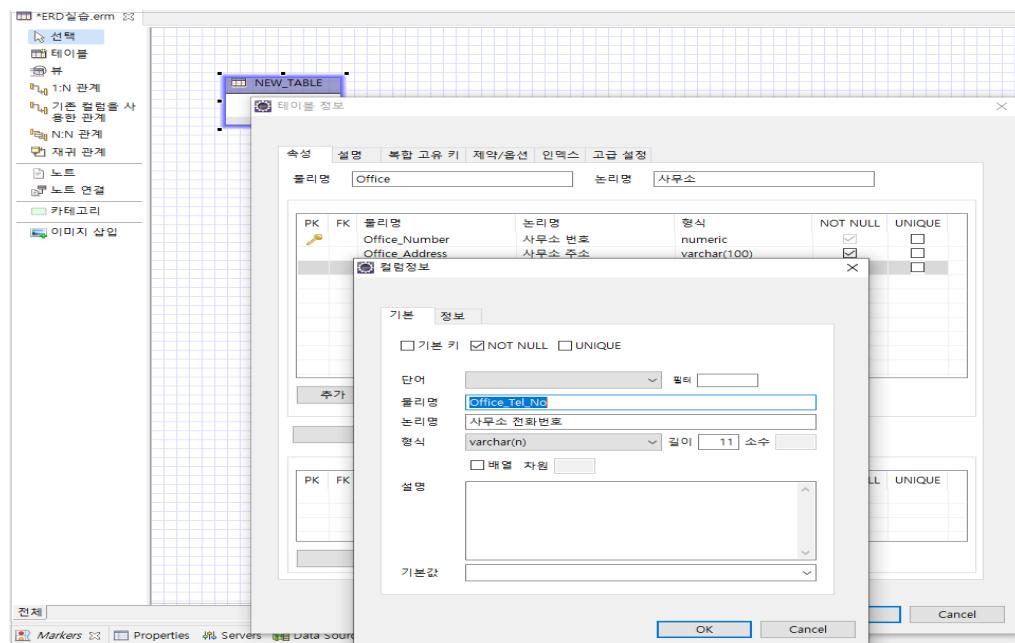


출처: 집필진 제작(2021)

[그림 3-13] E-R(Entity-Relationship) 다이어그램 작성 도구 실행

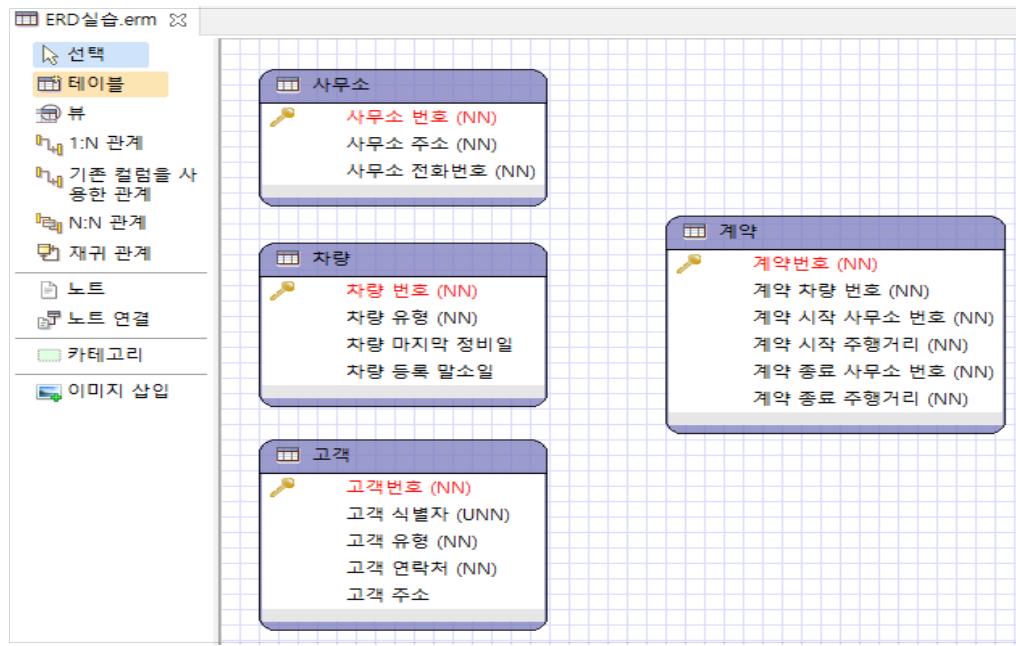
### (3) 개체를 작성하고 속성을 지정한다.

테이블을 선택하고 그리드를 클릭하여 생성된 테이블을 더블클릭하여 속성과 도메인을 입력한다.



출처: 집필진 제작(2021)

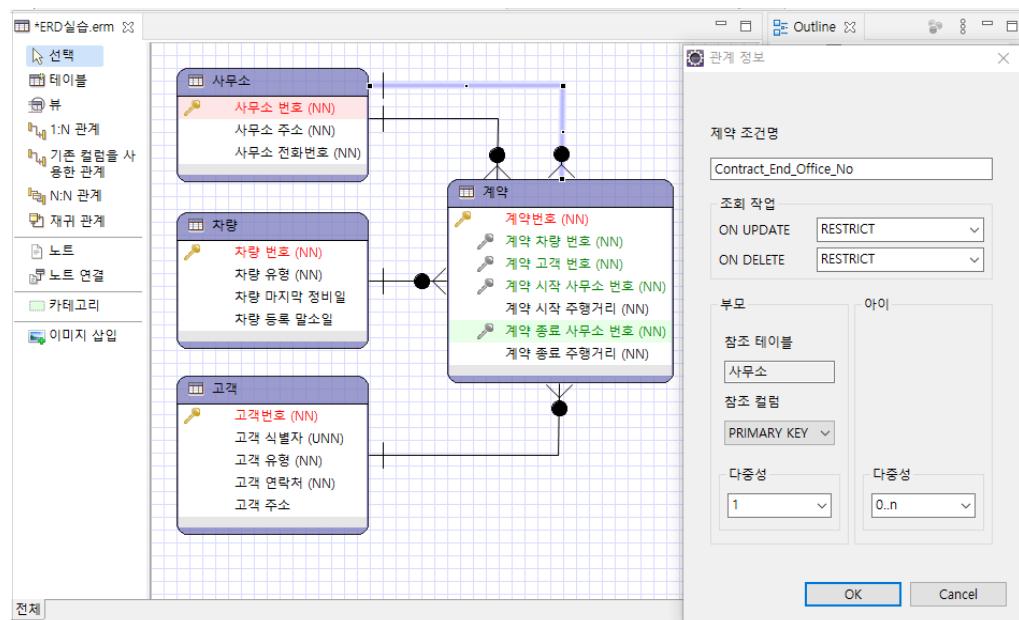
[그림 3-14] E-R(Entity-Relationship) 다이어그램 테이블 및 컬럼 추가



출처: 집필진 제작(2021)  
[그림 3-15] E-R(Entity-Relationship) 다이어그램 테이블 추가 예시

#### (4) 관계(Relationship)를 추가한다.

기존 컬럼을 사용한 관계를 클릭하여 각 테이블을 연결하고 관계선을 더블클릭하여 다중성 항목을 이용해 기수성, 선택성을 지정한다. 부모와 아이의 관계가 1:N이고 아이가 필수가 아닌 경우 부모의 다중성에 1을 입력하고 아이의 다중성에 0..n을 선택하여 기수성과 선택성을 지정할 수 있다. 아이가 필수가 아닌 경우 아이의 다중성에 0..n을 선택한다.



출처: 집필진 제작(2021)  
[그림 3-16] E-R(Entity-Relationship) 다이어그램 관계 추가 예시

## 정규화(Normalization), 반정규화(Denormalization)

데이터의 무결성을 확보하기 위해 논리 모델링 단계에서 정규화를 수행하고 물리 모델링 단계에서 반정규화를 수행한다. 이번 학습 예시는 단순하여 제외하였지만 추가 학습한다.

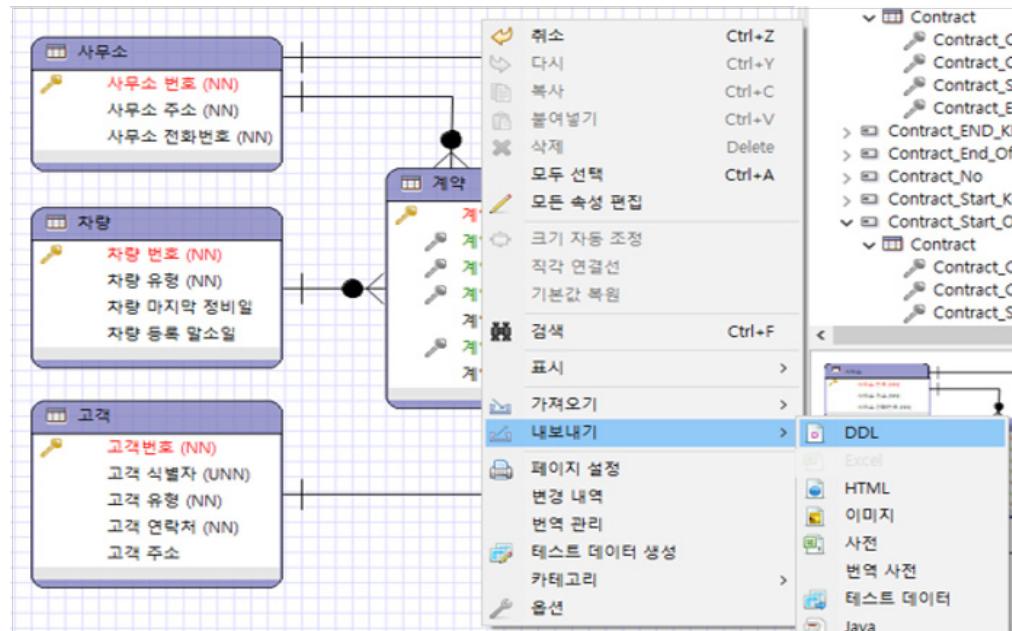
- 정규화: 데이터 중복 등으로 발생하는 삽입/삭제/갱신 이상현상(Anomaly)을 방지하기 위해 속성 간의 종속 관계를 제거하여 무손실 분해하는 기법
- 반정규화: 완전한 정규화를 수행한 이후에 성능, 관리 편의 등을 고려하여 테이블 또는 컬럼을 통합하는 기법

### ② 관계형 데이터베이스 테이블을 정의한다.

모델링 결과를 이용하여 테이블 정의서를 작성하고 데이터베이스에 적용하기 위한 쿼리(Query)를 생성한다. 작업 방법은 문서 작성 프로그램을 이용해 별도로 작성하는 것도 가능하지만 E-R(Entity-Relationship) 다이어그램 작성 도구가 제공하는 문서 변환 기능을 이용해서 작성하는 것도 가능하다. 도구를 이용해 변환하는 방법을 실습한다.

#### 1. 관계형 데이터베이스 테이블 생성 쿼리를 작성한다.

ERMaster 그리드에서 마우스 오른쪽 클릭하여 표시되는 내보내기에서 DDL을 클릭하고 위치를 지정하여 저장하면 테이블 생성 쿼리가 자동으로 생성된다.



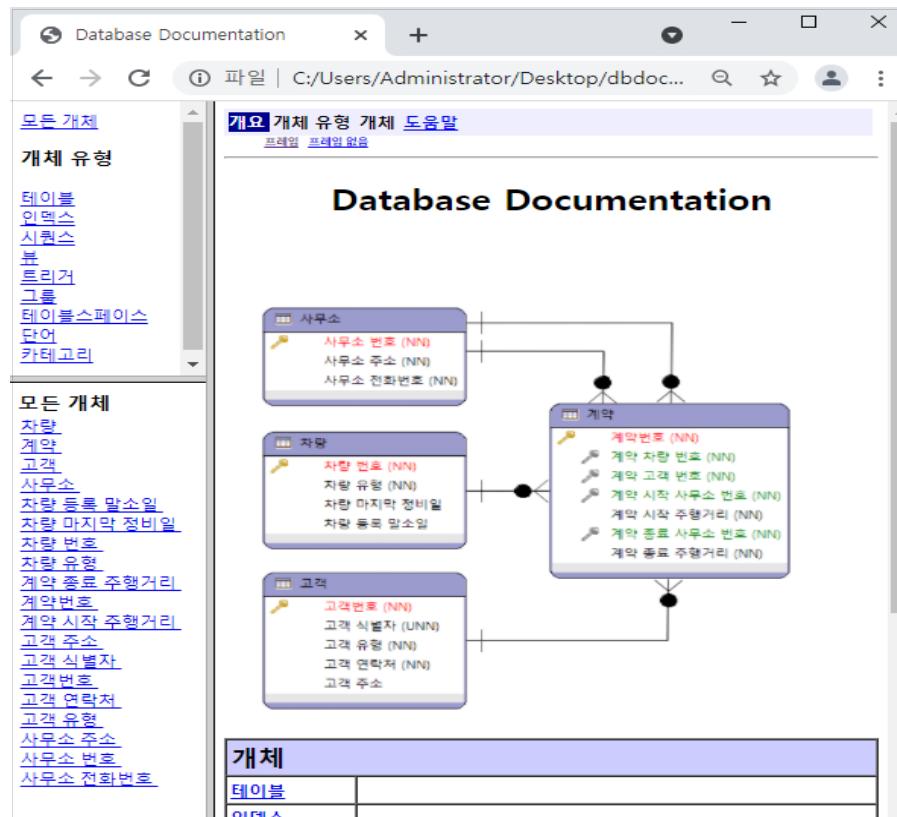
출처: 집필진 제작(2021)

[그림 3-17] 테이블 생성 쿼리 자동 생성 방법

#### 2. 관계형 데이터베이스 테이블 정의서를 생성한다.

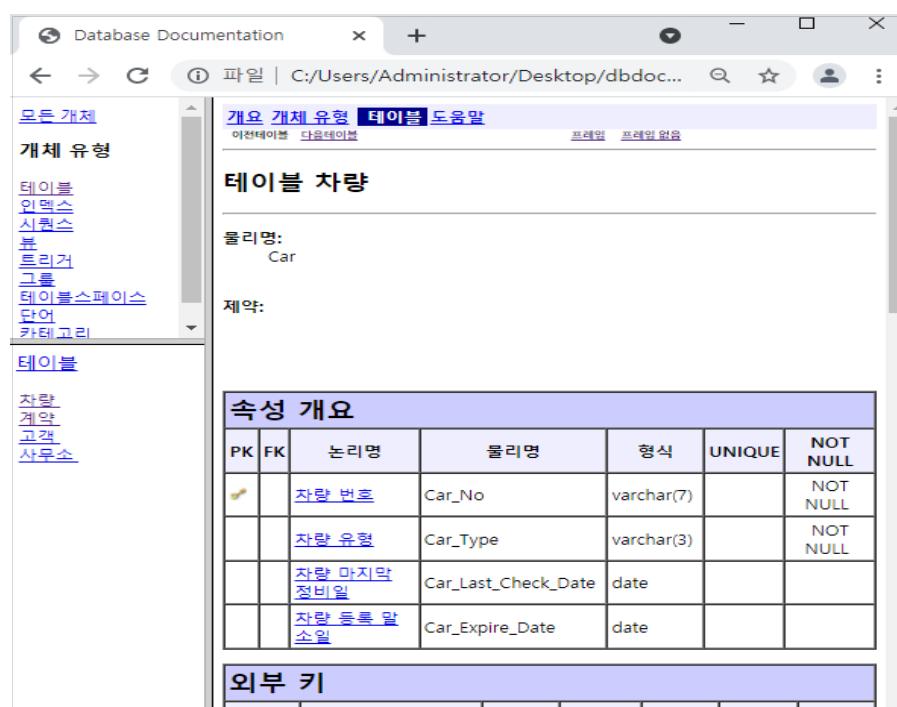
ERMaster 그리드에서 마우스 오른쪽 클릭하여 표시되는 내보내기에서 HTML 또는 문서 작성 프로그램을 클릭하고 위치를 지정하여 저장하면 아래 [그림 3-18]와 같이

E-R 다이어그램, 테이블 정의서 등이 자동으로 생성된다.



출처: 집필진 제작(2021)

[그림 3-18] 자동 생성된 E-R(Entity-Relationship) 다이어그램 예시



출처: 집필진 제작(2021)

[그림 3-19] 자동 생성된 테이블 정의서 예시

## 3-3. 관계형 데이터베이스 테이블 조작

### 학습 목표

- 테이블의 구조와 제약조건을 생성, 삭제하고 수정하는 DDL(Data Definition Language) 명령문을 작성할 수 있다.
- 한 개의 테이블에 대해 데이터를 삽입, 수정, 삭제하고 행을 조회하는 DML(Data Manipulation Language) 명령문을 작성할 수 있다.

### 필요 지식 /

#### ① SQL(Structured Query Language) 개요

##### 1. SQL 개념

SQL은 데이터베이스를 조작하기 위한 구조화된 질의어로 사용자가 처리 결과를 얻기 위한 질의어를 데이터베이스 관리시스템에 전달하면, 데이터베이스 관리시스템이 최적화 과정을 통해 처리 절차 및 방법을 수립하여 처리한다. 즉 사용자가 처리 과정을 직접적으로 고려할 필요 없는 비절차적 언어이다.

##### 2. SQL 표준

SQL은 ANSI(American National Standard Institute)와 ISO/IEC(International Standard Organization/International Electrotechnical Commission)에서 공동으로 표준화하여 대부분의 데이터베이스 관리시스템은 표준 SQL을 지원한다. 하지만 데이터베이스 관리시스템 솔루션들은 표준 SQL과 호환되지 않는 내장 함수 등을 제공하기 때문에 각 솔루션 고유의 SQL 문법을 고려하여 응용소프트웨어를 개발해야 한다.

〈표 3-20〉 표준 SQL 주요 버전

유형	설명
SQL-86	최초의 표준 SQL로 Select, Insert, Update, Delete, Group by, Having 등 지원
SQL-89	Unique, Not Null, Default, Check 등 무결성 제약 기능을 보강
SQL-92	현재 관계형 데이터베이스에서 사용되는 SQL의 기반이 되는 버전으로 SQL2라고도 함. Join이 추가되었으며 Case When, Cast 연산자 등이 추가됨
SQL:1999	SQL3라고도 하는 주요 개정으로 SQL의 기본 개념을 정의하고 기본 구문 및 연산을 정리했으며 Rollup, Cube 등 OLAP(Online Analytical Processing) 기능 추가됨
SQL:2003	2003년 원도 함수들이 추가되고 XML 관련 기능이 도입되었으며 이후부터는 정기적으로 업데이트되어 발표되고 있음. SQL:2016에는 JSON 지원도 추가됨

## ② SQL 구문

SQL 표준에서는 SQL 구문을 DDL(Data Definition Language, 데이터 정의어), DML(Data Manipulation Language, 데이터 조작어), DCL(Data Control Language, 데이터 제어어) 3가지 유형으로 정의하고 있다.

### 1. DDL(Data Definition Language, 데이터 정의어)

데이터를 정의하는 언어로 데이터베이스 스키마 구조를 변경하는 목적으로 데이터베이스 관리자나 설계자가 주로 사용한다.

〈표 3-21〉 DDL(Data Definition Language) 종류

종류	설명
CREATE	Database, Table, View, Index 등 생성
ALTER	Table의 속성(Attribute), 도메인(Domain), 제약조건(Constraint) 등 변경
DROP	Database, Table, View, Index 등 삭제
TRUNCATE	Table의 모든 행 삭제

### 2. DML(Data Manipulation Language, 데이터 조작어)

데이터를 조작하는 언어로 데이터를 조회하거나 입력, 수정, 삭제하기 위한 목적으로 데이터베이스 관리자나 응용소프트웨어 개발자가 주로 사용한다.

〈표 3-22〉 DML(Data Manipulation Language) 종류

종류	설명
SELECT	Table, View의 특정 조건 데이터 조회
INSERT	Table에 특정 조건의 데이터 입력
UPDATE	Table의 특정 조건 데이터 수정
MERGE	Table에 특정 조건에 따라 데이터를 입력(INSERT)하거나 수정(UPDATE)
DELETE	Table의 특정 조건의 데이터 삭제

### 3. DCL(Data Control Language, 데이터 제어어)

데이터를 제어하는 언어로 데이터베이스 접근을 위한 권한을 부여하거나 회수하는 목적으로 데이터베이스 관리자가 주로 사용한다.

〈표 3-23〉 DCL(Data Control Language) 종류

종류	설명
GRANT	특정 사용자에게 특정 객체에 대한 특정 작업 수행 권한 부여
REVOKE	특정 사용자에게 부여된 권한 회수

## 수행 내용 / 관계형 데이터베이스 테이블 조작하기

### 재료 · 자료

- E-R(Entity-Relationship) 다이어그램
- 테이블 정의서
- DDL, DML 실습 시나리오

### 기기(장비 · 공구)

- DBMS(Database Management System)
- 데이터베이스 개발 도구

### 안전 · 유의 사항

- DDL, DML 실습 시 반드시 쿼리 실행 후 결과를 확인하여 정상적으로 처리되었는지 확인한다.
- 관계형 데이터베이스 테이블 조작에 대한 기초 지식 함양을 목적으로 실습 환경을 고려하여 기초 기술을 실습하므로 응용소프트웨어 엔지니어링 실무에 활용하기 위해서는 실습 내용을 토대로 SQL 활용 능력에 대해 심화 학습한다.

### 수행 순서

#### ① DDL 명령문을 작성한다.

학습 내용 3-2의 수행 내용 ②에서 정의했던 차량, 계약 테이블을 이용해 아래 시나리오와 같이 테이블 구조와 제약조건을 생성, 삭제하고 수정하는 DDL 명령문을 작성한다.

#### DDL 실습 시나리오

1. 차량 번호를 주키(Primary Key)로 하여 차량 테이블을 생성하고, 계약 번호를 주키(Primary Key)로 하여 계약 테이블을 생성한다.
2. 계약 테이블의 차량 번호에 외래키(Foreign Key) 제약조건을 추가한다.
3. 계약 테이블에 계약일시 컬럼을 추가하고 Not Null 제약조건을 지정한다.
4. 차량 테이블과 계약 테이블에 데이터를 입력한다.
5. 차량 테이블을 삭제하고 오류가 발생하는 것을 확인한다.
6. 계약 테이블의 차량 번호 외래키에 ON DELETE CASCADE 옵션을 추가한다.
7. 차량 테이블을 삭제하고 계약 테이블의 데이터가 같이 삭제된 것을 확인한다.

1. 차량 번호를 주키(Primary Key)로 하여 차량 테이블을 생성하고 계약 번호를 주키

(Primary Key)로 하여 계약 테이블을 생성한다.

(1) 테이블 생성 SQL 문법을 확인한다.

#### 테이블 생성 SQL 문법

```
CREATE TABLE 테이블명 (
    컬럼명1 자료형 [DEFAULT 기본값] [제약조건],
    ....
    컬럼명N 자료형 [DEFAULT 기본값] [제약조건]
);
```

#### 수행 tip

- SQL 문법에서 대괄호 []은 생략 가능한 옵션 부분이다.
- 주키(Primary Key) 등 제약조건명을 입력하지 않으면 자동으로 지정된다.

(2) 테이블 생성 SQL을 작성하여 실행한 후 결과를 확인한다.

#### 테이블 생성 SQL 작성 사례

```
-- 차량 테이블 생성
CREATE TABLE CAR (
    CAR_NO VARCHAR(7) PRIMARY KEY,
    CAR_TYPE VARCHAR(3) NOT NULL,
    CAR_LAST_CHECK_DATE DATE,
    CAR_EXPIRE_DATE DATE );

-- 계약 테이블 생성
CREATE TABLE CONTRACT (
    CONTRACT_NO NUMERIC(10) PRIMARY KEY,
    CONTRACT_CAR_NO VARCHAR(7) not NULL,
    CONTRACT_CUSTOMER_NO NUMERIC(10) not NULL,
    CONTRACT_START_OFFICE_NO NUMERIC(10) not NULL,
    CONTRACT_START_KM NUMERIC(10) not NULL,
    CONTRACT_END_OFFICE_NO NUMERIC(10),
    CONTRACT_END_KM NUMERIC(10) );
```

2. 계약 테이블의 차량 번호에 외래키(Foreign Key) 제약조건을 추가한다.

(1) 외래키 추가 SQL 문법을 확인한다.

#### 외래키(Foreign Key) 추가 SQL 문법

```
ALTER TABLE 테이블명 ADD CONSTRAINT [제약조건명] FOREIGN KEY ([컬럼명])
    REFERENCES [참조 대상 테이블명]([참조 대상 컬럼명])
    [외래키 옵션];
```

## 외래키(Foreign Key) 추가 SQL 문법

외래키는 삭제 시 적용되는 ON DELETE 옵션과 수정 시 적용되는 ON UPDATE 옵션 존재

- 1) ON DELETE CASCADE: 부모 데이터 삭제 시 자식 데이터도 삭제
- 2) ON DELETE SET NULL: 부모 데이터 삭제 시 자식 데이터는 NULL로 수정
- 3) ON DELETE SET DEFAULT: 부모 데이터 삭제 시 자식 데이터는 DEFAULT값 수정
- 4) ON DELETE RESTRICT: 자식 테이블이 참조하고 있는 경우 데이터 삭제 불가
- 5) ON DELETE NO ACTION: RESTRICT와 동일하며 옵션 미지정 시 자동 선택됨
- 6) ON UPDATE 도 상기 1) ~ 5) 와 동일

(2) 외래키 추가 SQL을 작성하여 실행한 후 결과를 확인한다.

### 외래키 추가 SQL 작성 사례

```
ALTER TABLE CONTRACT
ADD CONSTRAINT CONTRACT_FK_1
FOREIGN KEY (CONTRACT_CAR_NO) REFERENCES CAR(CAR_NO);
```

3. 계약 테이블에 계약일시 컬럼을 추가하고 Not Null 제약조건을 지정한다.

(1) 컬럼 추가 SQL 문법을 확인한다.

## 컬럼 추가 SQL 문법

```
ALTER TABLE 테이블명 ADD 추가대상컬럼명 자료형 [DEFAULT 기본값] [제약조건];
```

(2) 컬럼 추가 SQL을 작성하여 실행한 후 결과를 확인한다.

### 컬럼 추가 SQL 작성 사례

```
ALTER TABLE CONTRACT ADD CONTRACT_DATE DATE NOT NULL;
```

4. 차량 테이블과 계약 테이블에 데이터를 입력한다.

(1) 데이터 입력 SQL 문법을 확인한다.

## 데이터 입력 SQL 문법

```
INSERT INTO 테이블명 [(컬럼명1, ... 컬럼명N)] VALUES(값1, ..., 값N);
```

(2) 데이터 입력 SQL을 작성하여 실행한 후 결과를 확인한다.

### 데이터 입력 SQL 작성 사례

```
-- 차량 테이블 데이터 입력
INSERT INTO CAR VALUES('64무1234','A',null, NULL);
-- 계약 테이블 데이터 입력
INSERT INTO CONTRACT
VALUES('2107180001', '64무1234', 1, 100, 0, null, null, NOW());
```

### 수행 tip

- 현재 시간을 취득하는 함수는 데이터베이스 관리시스템마다 다르다. 이번 실습은 PostgreSQL를 사용했기 때문에 NOW()를 사용한다.

5. 차량 테이블을 삭제하고 오류가 발생하는 것을 확인한다.

(1) 데이터 삭제 SQL 문법을 확인한다.

#### 데이터 삭제 SQL 문법

```
DELETE FROM 테이블명 [WHERE 조건식];
```

(2) 데이터 삭제 SQL을 작성하여 실행한 후 오류 발생을 확인한다.

#### 데이터 삭제 SQL 작성 사례

```
DELETE FROM CAR;
```

#### 데이터 삭제 SQL 오류 발생 사례

```
SQL Error [23503]: ERROR: update or delete on table "car" violates foreign key constraint "contract_fk_1" on table "contract"
```

```
Detail: Key (car_no)=(64무1234) is still referenced from table "contract".
```

### 수행 tip

- 외래키 생성 시 옵션을 지정하지 않으면 ON DELETE RESTRICT 옵션이 자동으로 지정되기 때문에 자식 테이블의 데이터가 남겨진 상태에서 부모 테이블의 데이터를 삭제하여 오류가 발생한 것을 확인한다.

6. 계약 테이블의 차량 번호 외래키에 ON DELETE CASCADE 옵션을 추가한다.

(1) 제약조건 삭제 SQL 문법을 확인한다.

#### 제약조건 삭제 SQL 문법

```
ALTER TABLE 테이블명 DROP CONSTRAINT [제약조건명]
```

(2) 외래키 삭제 후 옵션 추가 외래키 생성 SQL을 작성하여 실행한 후 결과를 확인한다.

#### 데이터 삭제 SQL 작성 사례

```
ALTER TABLE CONTRACT DROP CONSTRAINT CONTRACT_FK_1;  
ALTER TABLE CONTRACT  
ADD CONSTRAINT CONTRACT_FK_1  
FOREIGN KEY (CONTRACT_CAR_NO) REFERENCES CAR(CAR_NO)  
ON DELETE CASCADE;
```

7. 차량 테이블을 삭제하고 계약 테이블의 데이터가 같이 삭제된 것을 확인한다.

데이터 삭제 SQL을 작성하여 실행한 후 외래키의 ON DELETE CASCADE 옵션으로 인해 차량 테이블과 계약 테이블의 데이터가 같이 삭제된 것을 확인한다.

#### 데이터 삭제 SQL 작성 사례

```
DELETE FROM CAR;
```

## ② DML 명령문을 작성한다.

학습 내용 3-2의 수행 내용 ②에서 정의했던 차량 테이블을 이용해 아래 시나리오와 같이 한 개의 테이블에 대해 데이터를 삽입, 수정, 삭제하고 행을 조회하는 DML 명령문을 작성한다.

#### DDL 실습 시나리오

- 차량 테이블에 차량 유형이 A인 데이터를 2건, B인 데이터 2건, C인 데이터 2건을 입력한다.
- 차량 유형 B인 데이터의 차량 유형을 D로 수정한다.
- 차량 유형 A인 데이터를 삭제한다.
- 차량 유형 D인 데이터를 조회한다.

1. 차량 테이블에 차량 유형이 A인 데이터를 2건, B인 데이터 2건, C인 데이터 2건을 입력한다.

(1) 데이터 추가 SQL 문법을 확인한다.

#### 데이터 추가 SQL 문법

```
INSERT INTO 테이블명 [(컬럼명1,...컬럼명N)] VALUES (값1, ... 값N);
```

(2) 데이터 추가 SQL을 작성하여 실행한 후 결과를 확인한다.

#### 데이터 삭제 SQL 작성 사례

```
INSERT INTO CAR VALUES('64무0001','A',null, NULL);
INSERT INTO CAR(CAR_NO, CAR_TYPE) VALUES('64무0002','A');
INSERT INTO CAR(CAR_NO, CAR_TYPE) VALUES('64무0003','B');
INSERT INTO CAR(CAR_NO, CAR_TYPE) VALUES('64무0004','B');
INSERT INTO CAR(CAR_NO, CAR_TYPE) VALUES('64무0005','C');
INSERT INTO CAR(CAR_NO, CAR_TYPE) VALUES('64무0006','C');
```

#### 수행 tip

- 컬럼명 리스트를 지정하지 않으면 전체 컬럼이 대상이 되어 순차적으로 값 리스트를 지정해야 한다.
- 컬럼명 미지정 시 NULL 값이 자동 입력된다.

2. 차량 유형 B인 데이터의 차량 유형을 D로 수정한다.

(1) 데이터 수정 SQL 문법을 확인한다.

#### 데이터 수정 SQL 문법

```
UPDATE 테이블명 SET 수정컬럼명1 = 값1, ... 수정컬럼명N = 값N [WHERE 조건문];
```

(2) 데이터 수정 SQL을 작성하여 실행한 후 결과를 확인한다.

#### 데이터 수정 SQL 작성 사례

```
UPDATE CAR SET CAR_TYPE = 'D' WHERE CAR_TYPE = 'B';
```

3. 차량 유형 A인 데이터를 삭제한다.

(1) 데이터 삭제 SQL 문법을 확인한다.

#### 데이터 삭제 SQL 문법

```
DELETE [FROM] 테이블명 [WHERE 조건문];
```

(2) 데이터 삭제 SQL을 작성하여 실행한 후 결과를 확인한다.

#### 데이터 삭제 SQL 작성 사례

```
DELETE FROM CAR WHERE CAR_TYPE = 'A';
```

4. 차량 유형 D인 데이터를 조회한다.

(1) 데이터 조회 SQL 문법을 확인한다.

#### 데이터 조회 SQL 문법

```
SELECT (컬럼명1, ... 컬럼명N | *) FROM 테이블명 [WHERE 조건문];
```

#### 수행 tip

- 모든 컬럼을 조회하기 위해서는 컬럼명을 지정하지 않고 \*로 입력한다. 하지만 응용 프로그램을 개발할 때는 컬럼만 추가되고 응용소프트웨어는 변경하지 않아 오류가 발생될 수 있으니 \*로 조회하는 것은 지양해야 한다.

(2) 데이터 조회 SQL을 작성하여 실행한 후 결과를 확인한다.

#### 데이터 조회 SQL 작성 사례

```
SELECT CAR_NO, CAR_TYPE FROM CAR WHERE CAR_TYPE = 'D';
```

#### 교수 방법

- 데이터베이스 특성에 대해 충분히 설명하여 데이터베이스 특징을 식별하는 데 토대가 되도록 지도한다.
- E-R 다이어그램을 이해할 수 있도록 데이터 모델링에 대해 설명한다.
- 작성된 E-R 다이어그램을 E-R 다이어그램 도구를 이용해 테이블 정의서와 DDL 쿼리로 자동 변경하는 실습을 하여 능숙한 도구 활용의 효율성을 체험할 수 있도록 지도한다.
- SQL 표준에 대해 설명하고 다양한 DBMS 솔루션들의 SQL 표준 호환성을 이해하고 실습 할 수 있도록 지도한다.
- DDL과 DML의 차이를 설명하고 학습자 스스로 SQL 구문을 작성할 수 있도록 SQL의 기본 구조를 설명한다.

#### 학습 방법

- 데이터베이스 특성에 대해 정확히 이해하고 이를 토대로 데이터베이스 특징을 정리한다.
- 데이터 모델링을 사전 학습하고 데이터 모델링의 산출물인 E-R 다이어그램을 파악한다.
- E-R 다이어그램 도구를 이용해 테이블 정의서 작성 및 DDL 쿼리를 자동으로 변환시키는 과정을 실습한다.
- SQL 표준에 대해 학습하여 다양한 DBMS 솔루션들의 SQL 표준 호환성을 이해하고 솔루션 고유의 SQL 구문이 필요한 경우에는 추가로 조사하여 실습에 활용한다.
- DDL과 DML의 차이를 파악하고 SQL의 기본 구조를 이해하여 스스로 다양한 SQL 구문 을 작성한다.

## 학습3 평 가

### 평가 준거

- 평가자는 학습자가 학습 목표를 성공적으로 달성하였는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.

학습 내용	학습 목표	성취수준		
		상	중	하
데이터베이스 특징 식별	- 데이터베이스의 종류를 구분하고 응용소프트웨어 개발에 필요한 데이터베이스의 특징을 식별할 수 있다.			
관계형 데이터베이스 테이블 정의	- 주어진 E-R(Entity-Relationship) 다이어그램을 이용하여 관계형 데이터베이스의 테이블을 정의할 수 있다.			
관계형 데이터베이스 테이블 조작	- 테이블의 구조와 제약조건을 생성, 삭제하고 수정하는 DDL(Data Definition Language) 명령문을 작성할 수 있다.  - 한 개의 테이블에 대해 데이터를 삽입, 수정, 삭제하고 행을 조회하는 DML(Data Manipulation Language) 명령문을 작성할 수 있다.			

### 평가 방법

- 서술형 시험

학습 내용	평가 항목	성취수준		
		상	중	하
데이터베이스 특징 식별	- 데이터베이스 종류 파악 여부  - 데이터베이스 특징 숙지 여부			
관계형 데이터베이스 테이블 정의	- 데이터 모델링 개념 숙지 여부  - E-R 표기법 숙지 여부			
관계형 데이터베이스 테이블 조작	- 관계형 데이터베이스 테이블 구조 숙지 여부  - DDL 개념 및 종류 숙지 여부  - DML 개념 및 종류 숙지 여부			

• 작업장 평가

학습 내용	평가 항목	성취수준		
		상	중	하
관계형 데이터베이스 테이블 정의	- E-R 다이어그램 작성 능력			
	- E-R 다이어그램 도구 활용 능력			
	- 관계형 데이터베이스 테이블 정의서 작성 능력			
관계형 데이터베이스 테이블 조작	- 테이블 구조와 제약조건을 생성, 삭제, 수정하는 SQL 작성 능력			
	- 데이터를 삽입, 수정, 삭제, 조회하는 SQL 작성 능력			
	- 데이터베이스 개발 도구 사용 능력			

**피드백**

1. 서술형 시험

- 데이터베이스 종류를 파악하고 응용소프트웨어 개발을 위해 숙지해야 하는 데이터베이스 특징을 파악하고 있는지 평가하고, 미흡한 부분에 대해서는 토론을 통해 보완한다.
- 데이터 모델링, E-R 다이어그램, 관계형 테이블의 연관성을 이해하고 있는지 평가하고, 미흡한 부분에 대해서는 예시를 통해 이해하기 쉽도록 설명한다.
- E-R 다이어그램 표기법을 파악하고 있는지 평가하고 부족한 부분은 추가 설명한다.
- SQL 구문의 유형 및 각 유형별 상세 SQL 구문의 종류를 숙지하고 있는지 파악하고 미흡한 부분은 자료 조사를 통해 보완한다.

2. 작업장 평가

- E-R 다이어그램 도구를 이용해 E-R 다이어그램을 작성하는 수준을 평가하고, 성취수준이 낮은 학습자들은 모둠별로 같이 작성하면서 보완하고, 성취수준이 높은 학습자는 난이도를 높여 작성하도록 심화 설명한다.
- 데이터베이스 개발 도구를 이용해 SQL 구문을 작성하는 수준을 평가하고, 미흡한 부분은 예시를 통해 추가 설명하고 평가 결과가 높은 학습자는 난이도를 높여 작성하도록 심화 설명한다.

# 참고자료



- 고재성, 이상훈(2020). 『IT 엔지니어를 위한 네트워크 입문』. 길벗.
- 교육부(2020). 응용 SW기초 기술 활용(LM2001020216\_15v3). 한국직업능력개발원.
- 손진곤, 김준민(2015). 『정보통신망』. 한국방송통신대학교출판문화원.
- 이석호(2010). 『데이터베이스론』. 정의사.
- 한국데이터산업진흥원(2020). 『데이터아키텍처 전문가 가이드』. 한국데이터산업진흥원.
- 행정안전부(2016). 표준프레임워크 소개. <https://www.egovframe.go.kr> 에서 2021. 07. 03. 검색.
- 행정안전부(2016). 표준프레임워크 구성. <https://www.egovframe.go.kr> 에서 2021. 07. 03. 검색.
- Apache Tomcat. User Guide. <https://tomcat.apache.org>에서 2021. 07. 03. 검색.
- ERMaster(<http://ermaster.sourceforge.net/update-site/>).
- Gene([2012] 2017). 『손으로 익히며 배우는 네트워크 첫걸음(おうちで学べるネットワークのきほん)』. 진솔(역). 한빛미디어.
- Hironobu SUZUKI(2021). Fig.2.1 An example of the process architecture. [https://www.interdb.jp/pg/pgsql02.html#\\_2.1](https://www.interdb.jp/pg/pgsql02.html#_2.1).에서 2021. 07. 03 검색.
- Hironobu SUZUKI(2021). Fig.3.2 An example of a parse tree. <https://www.interdb.jp/pg/pgsql03.html>에서 2021. 07. 03. 검색.
- Hironobu SUZUKI(2021). Fig.3.5 A simple plan tree. <https://www.interdb.jp/pg/pgsql03.html>에서 2021. 07. 03. 검색.
- OnopenJDK. Download. <https://openjdk.java.net>에서 2021. 07. 03. 검색.
- PostgreSQL. Download. <https://www.postgresql.org>에서 2021. 07. 03. 검색.
- SW중심사회(2017). 글로벌 상용SW 백서 미들웨어SW. <http://www.software.kr>에서 2021. 07. 03. 검색.
- Wireshark. Download. <https://www.wireshark.org>에서 2021. 07. 03. 검색.

## NCS학습모듈 개발이력

발행일	2021년 12월 31일
세분류명	응용SW엔지니어링(20010202)
개발기관	(사)한국정보통신기술사협회(개발책임자: 온기현), 한국직업능력연구원
집필진	김승환(주캐롯아이)* 김원기(LG CNS) 박주형((주)오아시스비즈니스) 엄기영(우리FIS) 윤혜경(만도) 최홍선((주)에이스기술단)
검토진	안응원(미라콤아이앤씨) 유현주(한국전산감리원) 홍필두(한국폴리텍대학)
	*표시는 대표집필자임

## 응용SW 기초 기술 활용(LM2001020232\_19v4)

저작권자	교육부
연구기관	한국직업능력연구원
발행일	2021. 12. 31.
ISBN	979-11-339-9445-8

※ 이 학습모듈은 자격기본법 시행령(제8조 국가직무능력표준의 활용)에 의거하여 개발하였으며,  
NCS통합포털사이트(<http://www.ncs.go.kr>)에서 다운로드 할 수 있습니다.



[www.ncs.go.kr](http://www.ncs.go.kr)