

WEB PROGRAMMING

01/31 14:00 PM

JSP Action Tag

Action Tag

- ✓ JSP 페이지 내에서 어떤 동작을 하도록 지시하는 태그
- ✓ 종류 : forward, include, param, bean 등...
- ✓ <jsp:사용할 속성> </jsp:사용할 속성>
 - ✓ 태그 내부에 작성할 내용 없다면 닫는 태그 사용 X, 열린 태그 마지막에 / 삽입

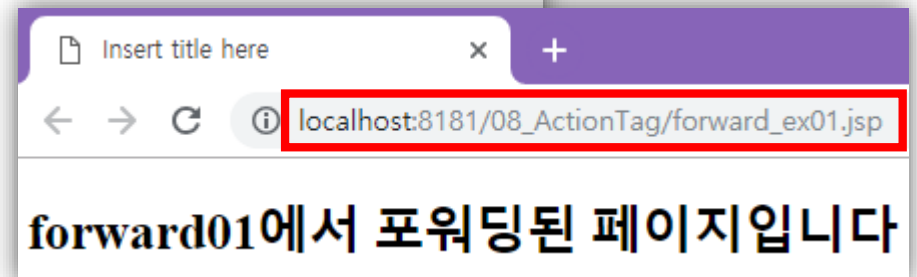
Forward

- ✓ 현재 페이지에서 다른 특정 페이지로 전환할 때 사용
- ✓ `<jsp:forward page="페이지 URL / 디렉토리"> </jsp:forward/>`
- ✓ `sendRedirect()`와 같은 기능 BUT 그와 다르게 리다이렉트된 페이지 주소가 아니라 해당 페이지 주소가 뜸
- ✓ forward의 원형 클래스는 `RequestDispatcher` 클래스

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!-- action 태그 사용법 <jsp:사용할 속성></jsp:사용할 속성>-->
```

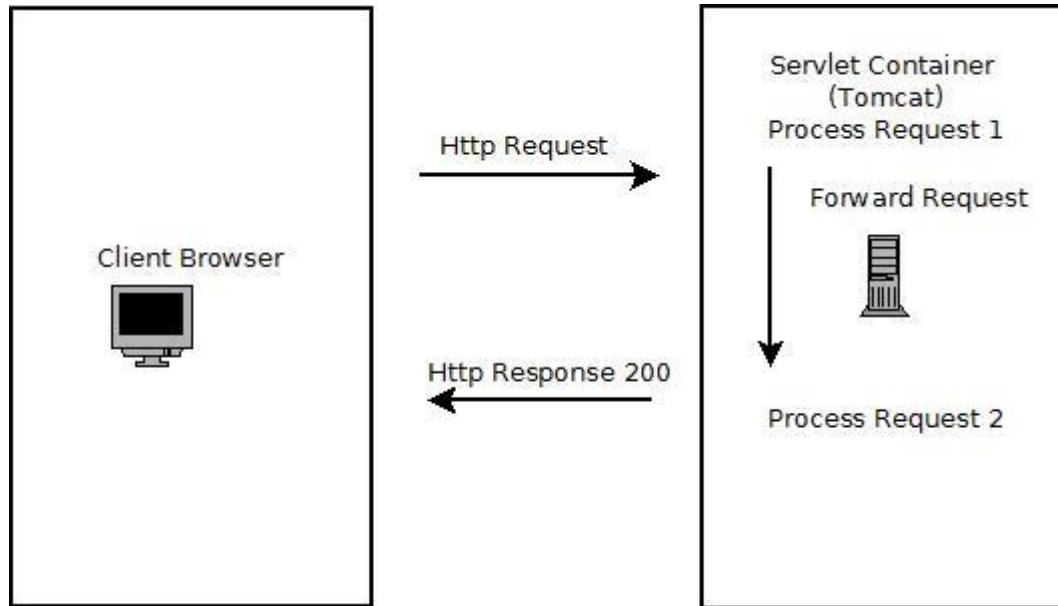
```
<jsp:forward page = "forward_ex02.jsp"/>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
    <h1>forward01 페이지입니다.</h1>
    <h2>페이지 강제 이동을 하면 이 페이지가 나타날까요?</h2>
</body>
</html>
```

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
    <h1>forward01에서 포워딩된 페이지입니다</h1>
</body>
</html>
```



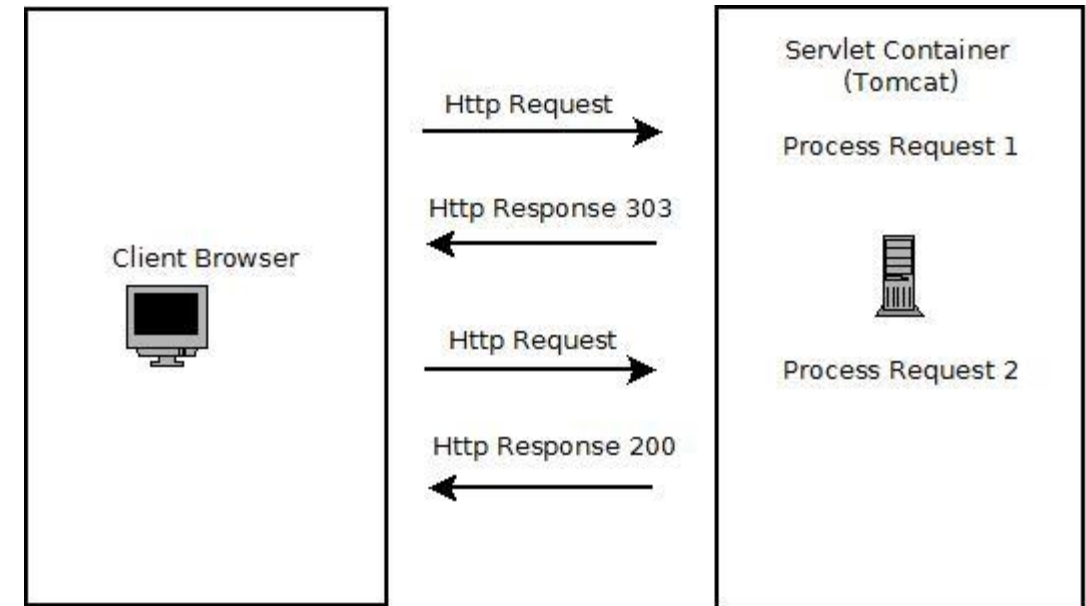
Forward vs Redirect

Forward



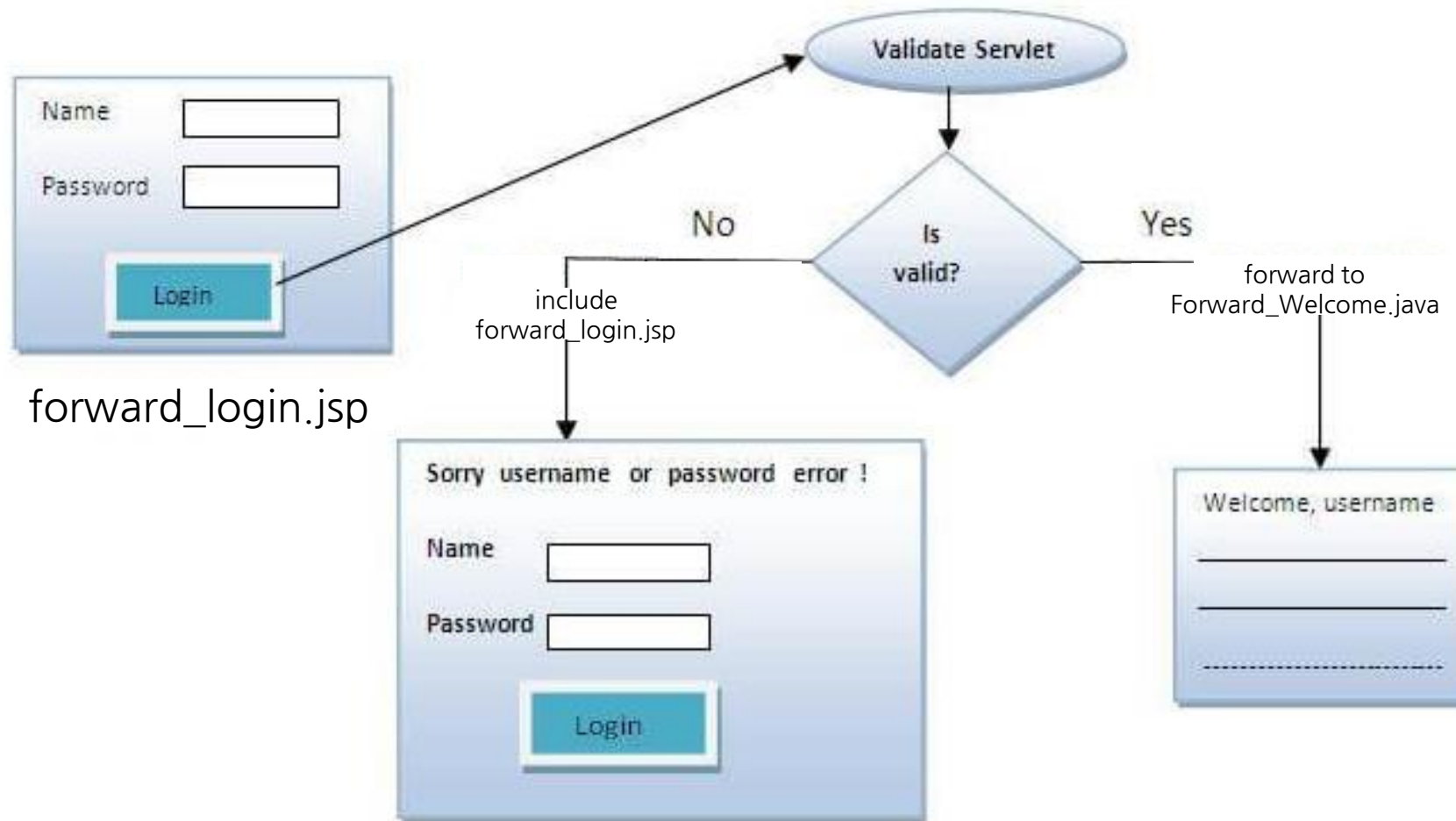
- 요청받은 Request 객체를 위임하는 컴포넌트에 요청 객체 값을 동일하게 전달 가능
- 요청 처리와 동시에 해당 결과를 바로 보여 줘야 하는 경우 사용
- ex) 게시판 목록에서 글 제목을 클릭했을 때 바로 내용을 보여 줘야 하는 경우

Redirect



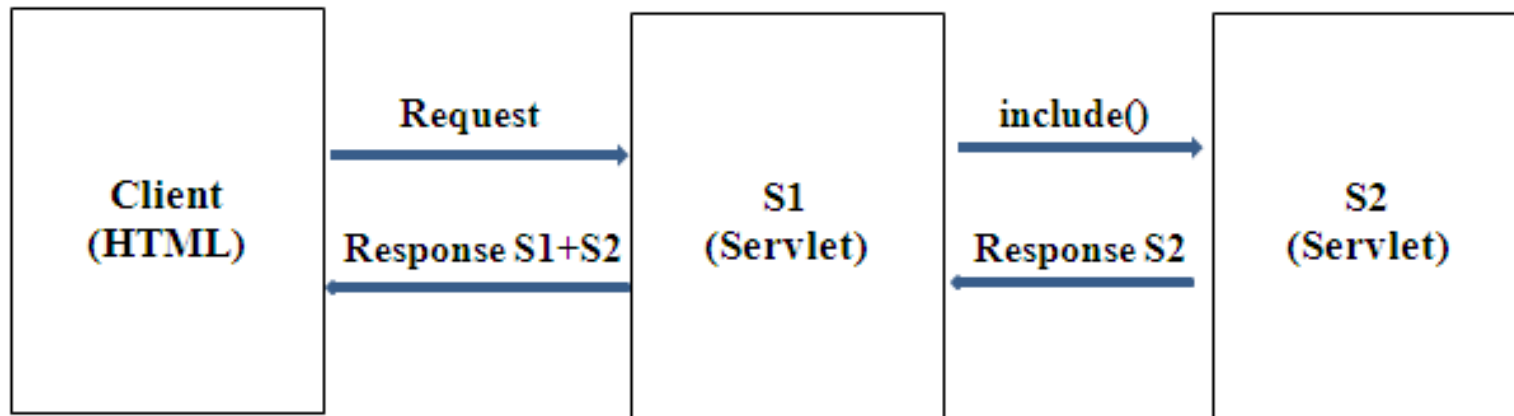
- 요청받은 Request 객체를 위임하는 컴포넌트에 전달하는 것이 아닌, 새로운 Request 객체를 생성
- 요청을 처리한 다음, 새로운 요청으로 작업을 해야 할 경우에 사용
- ex) 게시판 글 쓰기 완료 이후 해당 글의 내용이 아닌 글 목록을 보여줘야 하는 경우

Forward 사용한 로그인 구현



Include

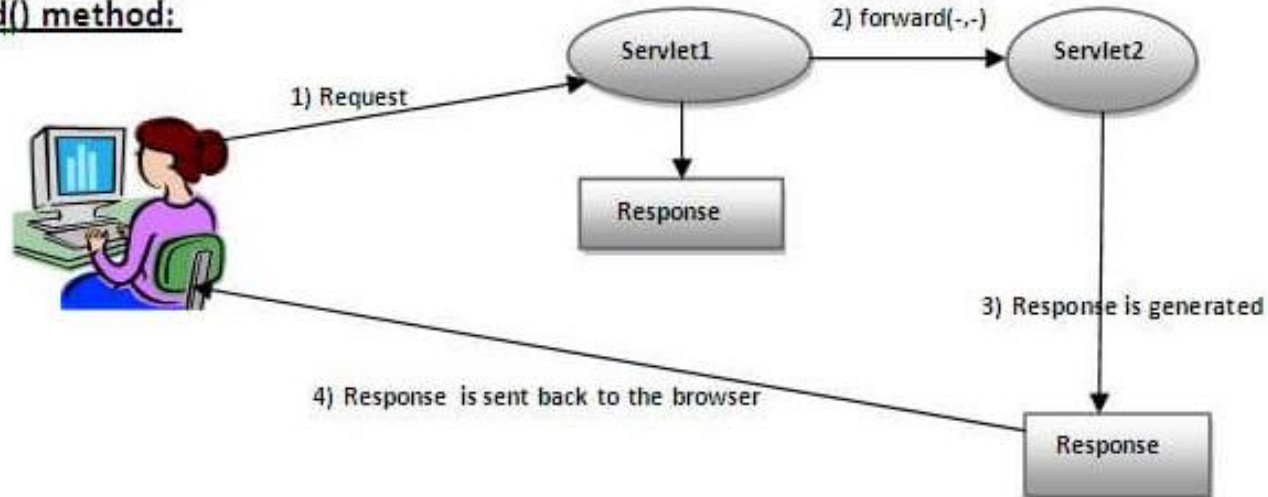
- ✓ 현재 페이지에 다른 페이지를 삽입할 때 사용
- ✓ `<jsp:include page="페이지 URL / 디렉토리"/>`
- ✓ 단순히 UI(디자인)만 복사하여 적용시키고 싶을 때 사용
- ✓ 페이지 디렉티브 `<%@ include file = "삽입할 페이지" %>`의 include 기능은 페이지 전역에 사용할 변수와 메소드가 있을 때 사용



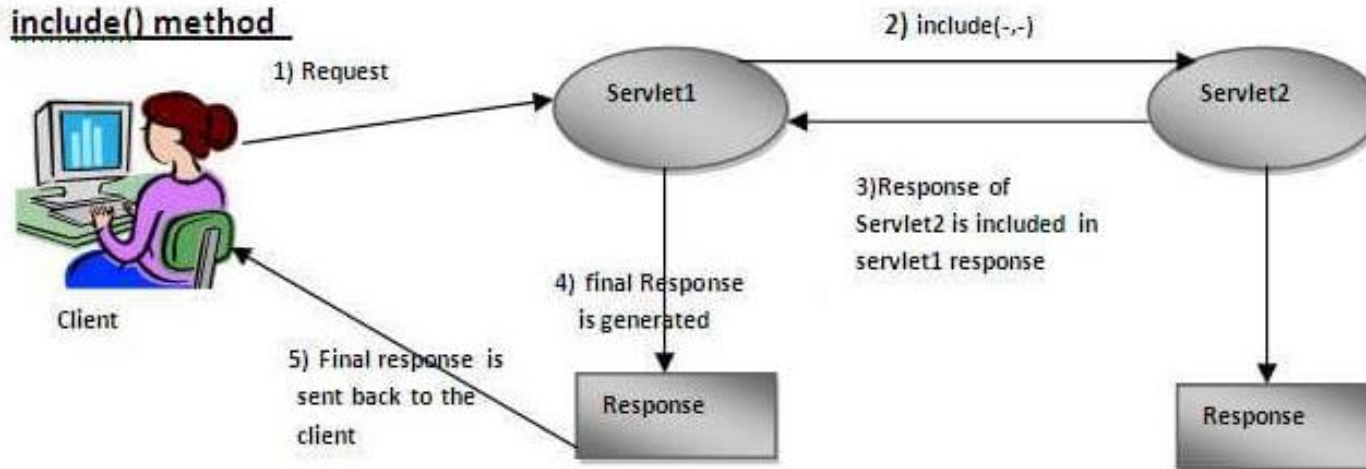
RequestDispatcher – include() method

Forward vs Include

forward() method:



include() method



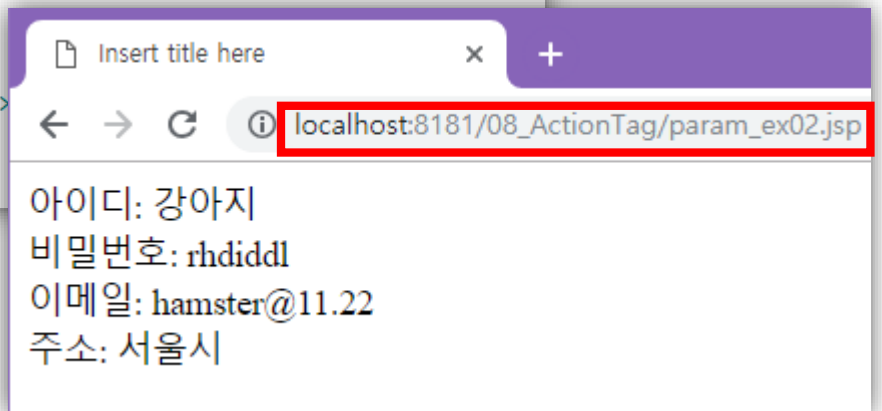
Param

- ✓ forward 및 include 태그에 데이터를 전달할 목적으로 사용되는 태그
- ✓ name과 value 속성으로 이루어져 있음
- ✓ `<jsp:param name = "데이터 이름" value = " 데이터 값"/>`

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
    <form action = "param_ex02.jsp" method = "post">
        아이디: <input type = "text" name = "id" size = "5"><br>
        비밀번호: <input type = "password" name = "pw" size = "10"><br>
        이메일: <input type = "text" name = "email" size = "10"><br>
        <input type = "submit" value = "확인">
    </form>
</body>
</html>
```

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
    String id = request.getParameter("id");
    String pw = request.getParameter("pw");
    String email = request.getParameter("email");
    String addr = request.getParameter("addr");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
    아이디: <%= id %><br>
    비밀번호: <%= pw %><br>
    이메일: <%= email %><br>
    주소: <%= addr %><br>
</body>
</html>
```

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<% request.setCharacterEncoding("utf-8"); %>
<jsp:forward page="param_ex03.jsp">
    <jsp:param name = "addr" value = "서울시"/>
</jsp:forward>
```



Java Bean

Java Bean

- ✓ Java 언어의 변수와 기능으로 이루어진 클래스
- ✓ 데이터 저장하는 변수, 데이터 읽어오는 getter 메소드, 데이터 저장할 때 사용하는 setter 메소드로 이루어짐
- ✓ DB와의 반복적인 작업을 효율적으로 처리하기 위해 사용
- ✓ JSP에서는 Action Tag를 사용하여 자바빈을 사용

```
<jsp:useBean id = "" class = "" scope = ""/>
```

- id - JSP 페이지에서 자바빈 객체에 접근할 때 사용할 이름을 지정
- class - 패키지 이름을 포함한 자바빈 클래스의 완전한 실제 경로를 입력
- scope - 자바빈 객체를 저장할 영역
 - page : 하나의 JS P페이지를 처리할 때 사용되는 영역
 - request : 하나의 요청을 처리할 때 사용되는 영역
 - session : 하나의 웹 브라우저와 관련된 영역
 - application : 하나의 웹 어플리케이션과 관련된 영역

Java Bean

✓ Getter 사용 방법

```
<jsp:setProperty name = "" property = "" value = "">
```

- name : 값을 변경할 자바빈 객체의 이름 (useBean 태그에서 id 속성에 지정한 값)
- property : 값을 지정할 프로퍼티의 이름 (자바빈 클래스의 변수명)
- value : 프로퍼티의 값을 지정 (표현식이나 EL도 사용 가능)

✓ Setter 사용 방법

```
<jsp:setProperty name = "" property = ""/>
```

- name : useBean 태그에서 id 속성에 지정한 값
- property : 출력할 프로퍼티의 이름을 지정 (자바빈 클래스의 변수명을 적음)

실습

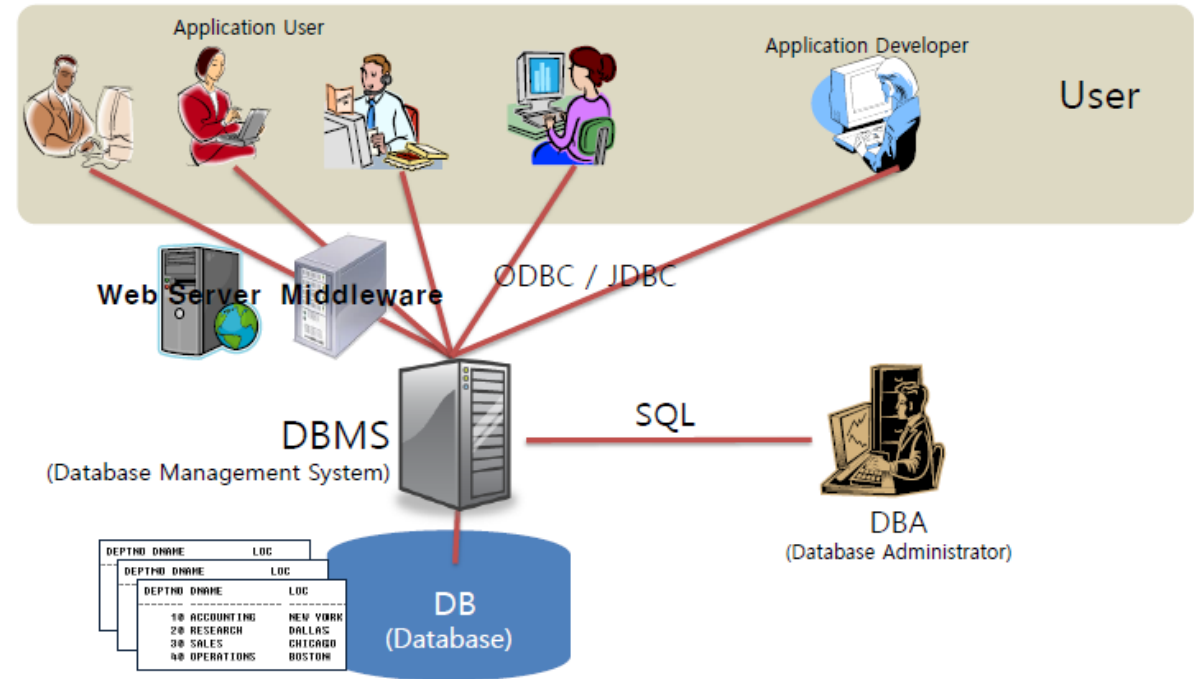
Database

Database

- 한 조직의 여러 응용 시스템들이 공용(Shared)하기 위해 통합(Integrated), 저장(Stored)한 운영 데이터(Operational data)의 집합
 - ✓ 컴퓨터 시스템과 무관하게 작동
 - ✓ 데이터의 추가, 삭제, 검색, 수정, 이동 등의 기능
 - ✓ 데이터 모델에 따라 데이터베이스 구조 달라짐
 - ✓ 이 데이터베이스를 관리하는 도구가 DBMS(DataBase Management System)

Database Management System (DBMS)

- DB 관리를 위한 컴퓨터 시스템
- ✓ SQL 언어와 DB 연결해 주는 도구
- ✓ 전사적인 정보 관리
- ✓ 데이터에 접근하는 프로그램 집합
- ✓ 효율적이고 편한 사용을 위한 환경
- ✓ 대표적으로 RDBMS (관계형 데이터베이스) 사용 多
- ✓ ex) Oracle, MySQL, MSSQL 등



Relational Model

- ✓ Table (relation) : set of tuples
- ✓ Tuple (row / record) : set of attributes' values
- ✓ Attribute : column
- ✓ Domain : attribute가 가질 수 있는 값의 집합

Student



학번	이름	연락처
1	김명명	010-1234-5678
2	박냥냥	010-9999-9999
3	이찍찍	010-8765-4321

attribute (column)

tuple (row / record)

domain = {1, 2, 3, 4} (학년)

SQL (Structured Query Language)

- SQL (Structured Query Language) : 데이터베이스의 데이터에 대해 작업을 수행할 때 사용하는 언어

➤ DDL (Data Definition Language)

- ✓ CREATE TABLE : 테이블이나 인덱스, 뷰 등 데이터베이스 객체 생성
- ✓ DROP TABLE : 생성된 데이터베이스 객체 삭제
- ✓ ALTER TABLE : 이미 생성된 데이터베이스 객체 수정
- ✓ TRUNCATE : 테이블이나 클러스터의 데이터를 통째로 삭제

➤ DML (Data Manipulation Language)

- ✓ SELECT : 테이블이나 뷰에 있는 데이터 조회
- ✓ INSERT : 데이터를 신규로 생성
- ✓ UPDATE : 이미 생성된 데이터를 수정
- ✓ DELETE : 데이터를 삭제
- ✓ COMMIT : 트랜잭션 처리, 변경된 데이터를 최종 적용
- ✓ ROLLBACK : 트랜잭션 처리, 변경된 데이터를 적용하지 않고 이전으로 되돌림

➤ DCL (Data Control Language)

- ✓ GRANT : 사용자에게 특정 권한을 부여
- ✓ REVOKE : 사용자에게 부여된 권한을 회수

SQL (Structured Query Language)

- SQL (Structured Query Language) : 데이터베이스의 데이터에 대해 작업을 수행할 때 사용하는 언어

SQL (Structured Query Language)

- SQL (Structured Query Language) : 데이터베이스의 데이터에 대해 작업을 수행할 때 사용하는 언어

실습