[ 7 ] 효율적인 데이터베이스 관리 – 회원인증 프로그램 예제

DTO, DAO, 커넥션풀

1. DAO, DTO
   * DAO : Data Access Object
     + 데이터 베이스에 접속해서 데이터 추가, 삭제, 수정 등의 작업을 하는 클래스이다
     + 일반적인 JSP 혹은 Servlet 페이지 내에 위의 로직을 함께 기술할 수도 있지만, 유지보수 및 코드의 모듈화를 위해 별도의 DAO클래스를 만들어 사용한다
   * DTO : Data Transfer Object
     + DAO 클래스를 이용하여 데이터 베이스에서 데이터를 관리할 때 데이터를 일반적인 변수에 할당하여 작업 할 수도 있지만, 해당 데이터의 클래스 객체를 만들어 사용한다. 데이터만 하나로 모아 객체에 담는다.

WAS

웹브라우저

데이터베이스

Servlet/JSP

Servlet/JSP

WAS

DTO

웹브라우저

데이터베이스

Servlet/JSP

DAO

DTO

1. 커넥션 풀
   * 클라이언트에서 다수의 요청이 발생할 경우 데이터베이스에 부하가 발생하게 된다

웹브라우저

WAS

DTO

데이터베이스

Servlet JSP

DAO

웹브라우저

DTO

웹브라우저

…

ⓒⓒ.. ⓒ

웹브라우저

* + 이러한 문제를 해결 하기 위해서 커넥션 풀(DataBase Connection Pool)기법을 이용

tomcat컨테이너가 데이터베이스 인증을 하도록 context.xml 파일에 아래의 코드를 추가

<Resource

auth=*"Container"*

driverClassName = *"oracle.jdbc.driver.OracleDriver"*

url = *"jdbc:oracle:thin:@localhost:1521:xe"*

username = *"scott"*

password = *"tiger"*

name = *"jdbc/Oracle11g"*

type = *"javax.sql.DataSource"*

maxActive = *"10"*

maxWait = *"1000"*

/>

</Context>

**커넥션 풀을 이용한 DAO**

**package** com.lec.member;

**import** java.sql.Connection;

**import** java.sql.Date;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.Timestamp;

**import** javax.naming.Context;

**import** javax.naming.InitialContext;

**import** javax.naming.NamingException;

**import** javax.sql.DataSource;

**public** **class** EmpDao {

**public** **static** EmpDao *instance*; //자기가 자기 클래스를 참조하고 있는 변수

**public** **static** EmpDao getInstance() {

**if**(*instance*==**null**) {

*instance* = **new** EmpDao();

}

**return** *instance*;

}

// conn 객체 가져오는 함수

**public** **static** Connection getConnection() **throws** Exception{

Connection conn = **null**;

**try** {

Context ctx = **new** InitialContext();

DataSource ds = (DataSource) ctx.lookup("java:comp/env/jdbc/Oracle11g");

conn = ds.getConnection();

} **catch** (NamingException e) {

System.***out***.println(e.getMessage());

}

**return** conn;

}

}

응용예제. 회원인증(customer).pdf 참조

통합구현 평가 : 친구 추가가 가능한 아래의 화면을 구현하시오(예시화면 참조) 120분

**<세부 요구사항>**

▪ 테이블 FRIEND(NO, NAME, TEL) 생성

▪ 시퀀스 FRIEND\_NO\_SEQ(1부터 하나씩 증가하는 시퀀스) 생성

▪ 친구 추가 및 친구 전체 목록 출력하는 QUERY 작성

▪ DTO 및 DAO(커넥션 풀 이용) 구현

▪ 뷰단 작성(friendInputList.jsp) : 아래 예시 화면과 같이 FRIEND 테이블에 있는 전체 친구 출력

▪ 프로세스단 작석ㅇ(friendInputListPro.jsp) : 파라미터로 넘어온 친구이름과 전화를 FRIEND테이블에 INSERT한 후 친구 추가 성공 여부를 경고창으로 뿌려주고 뷰단으로 이동한다.





**<파일명>**

|  |
| --- |
| friend.sql  FriendDto.java  FriendDato.java  friendInputList.jsp(뷰단)  friendInputListPro.jsp(프로세스단)  첫실행화면.png  추가실행화면.png |

**<결과물 제출>**

위의 파일이 압축된 파일 : 훈련생성명\_통합구현평가.zip

<참고사항>

- 테이블 FRIEND : no, name, tel

- 시퀀스 FRIEND\_NO\_SEQ 1부터 하나씩 증가하는 시퀀스

1. 친구 추가 QUERY

2. 친구 목록 출력 QUERY

- DTO, DAO(컨넥션 풀 이용)

- friendInputList.jsp (첫화면 = 뷰단)

- friendInputListPro.jsp(친구 추가 프로세스단)

**<평가항목>**

▪ 기능에 필요한 테이블 및 시퀀스를 올바로 생성

▪ 기능에 맞는 쿼리의 사용

▪ 효율적인 DB 접근에 필요한 DTO 구현

▪ 효율적인 DB 연결을 위한 커넥션 풀을 이용한 구현

▪ DAO 클래스의 싱글톤 구현

▪ DAO에서 쿼리에 맞는 값 반환 함수 사용

▪ 뷰단에서의 form구현과 FRIEND 테이블 데이터 출력

▪ 프로세스단에서 받은 파라미터를 DAO를 이용하여 insert할 함수를 이용한 구현

▪ 프로세스단에서 상황에 따른 경고창을 출력하고 뷰단으로 location 변환 구현

위의 구현을 제출후 다음의 추가 요구사항을 추가 할 것을 사색해 봅시다

