

〈11주차 실습〉

# PL/SQL(3)

---

Database Programming

 수강신청 시스템 : 프로시저, 함수, 트리거

 데이터베이스 어플리케이션 프로그래밍

 실습 과제

# **프로시저, 함수, 트리거 실습 : 수강신청 시스템**

# STUDENT TABLE

4/37

- **STUDENT**

STUDENT			
Column Name	Type	Key & Null Status	Remarks
S_ID	Varchar2(10)	PRIMARY KEY	
S_NAME	Varchar2(50)	NOT NULL	
S_ADDR	Varchar2(200)		
S_YEAR	NUMBER(1)	NOT NULL	
S_COLLEGE	Varchar2(50)	NOT NULL	
S_MAJOR	Varchar2(50)	NOT NULL	
S_PWD	Varchar2(10)	NOT NULL	

- **COURSE**

COURSE			
Column Name	Type	Key & Null Status	Remarks
C_ID	Varchar2(10)	PRIMARY KEY	
C_ID_NO	NUMBER(1)	PRIMARY KEY	
C_NAME	Varchar2(50)		
C_UNIT	NUMBER(1)		

# PROFESSOR TABLE

6/37

- PROFESSOR

PROFESSOR			
Column Name	Type	Key & Null Status	Remarks
P_ID	Varchar2(10)	PRIMARY KEY	
P_NAME	Varchar2(50)	NOT NULL	
P_COLLEGE	Varchar2(50)	NOT NULL	
P_MAJOR	Varchar2(50)	NOT NULL	
P_PWD	Varchar2(10)	NOT NULL	

- ENROLL

ENROLL			
Column Name	Type	Key & Null Status	Remarks
S_ID	Varchar2(10)	PRIMARY KEY	
C_ID	Varchar2(10)	PRIMARY KEY, FOREIGN KEY <Course(C_id)>	
C_ID_NO	NUMBER(1)	PRIMARY KEY, FOREIGN KEY <Course(C_id_no)>	
E_YEAR	NUMBER(4)		
E_SEMESTER	NUMBER(1)		
E_SCORE	NUMBER(3)		

- TEACH

TEACH			
Column Name	Type	Key & Null Status	Remarks
P_ID	Varchar2(10)	PRIMARY KEY	
C_ID	Varchar2(10)	PRIMARY KEY, FOREIGN KEY <Course(C_id)>	
C_ID_NO	NUMBER(1)	PRIMARY KEY, FOREIGN KEY <Course(C_id_no)>	
T_YEAR	NUMBER(4)	PRIMARY KEY	
T_SEMESTER	NUMBER(1)	PRIMARY KEY	
T_TIME	NUMBER(1)		
T_WHERE	Varchar2(50)		
T_MAX	NUMBER(2)		



```
SQL> CREATE OR REPLACE FUNCTION Date2EnrollYear(dDate IN DATE)
  2  RETURN NUMBER
  3  IS
  4      nYear NUMBER;
  5      sMonth CHAR(2);
  6  BEGIN
  7      /* 11월~4월: 1학기, 5~10월: 2학기 */
  8      SELECT TO_NUMBER(TO_CHAR(dDate, 'YYYY')), TO_CHAR(dDate, 'MM')
  9      INTO nYear, sMonth
 10      FROM DUAL;
 11      IF (sMonth = '11' or sMonth = '12') THEN
 12          nYear := nYear + 1;
 13      END IF;
 14      RETURN nYear;
 15  END;
 16  /
```

함수 사용

함수가 생성되었습니다.

```
SQL> CREATE OR REPLACE FUNCTION Date2EnrollSemester(dDate IN DATE)
  2  RETURN NUMBER
  3  IS
  4      nSemester NUMBER;
  5      sMonth CHAR(2);
  6  BEGIN
  7      /* 11월 ~ 4월 : 1학기,   5월 ~ 10월   : 2학기   */
  8      SELECT TO_CHAR(dDate, 'MM')
  9      INTO sMonth
 10      FROM DUAL;
 11      IF (sMonth='11' or sMonth='12' or (sMonth >='01' and sMonth<='04')) THEN
 12          nSemester := 1;
 13      ELSE
 14          nSemester := 2;
 15      END IF;
 16      RETURN nSemester;
 17  END;
 18  /
```

함수가 생성되었습니다.

```
SQL> CREATE OR REPLACE PROCEDURE InsertEnroll(sStudentId IN VARCHAR2,
2      sCourseId IN VARCHAR2,
3      nCourseIdNo IN NUMBER,
4      result OUT VARCHAR2)
5  IS
6      too_many_sumCourseUnit EXCEPTION;
7      too_many_courses EXCEPTION;
8      too_many_students EXCEPTION;
9      duplicate_time EXCEPTION;
10     nYear NUMBER;
11     nSemester NUMBER;
12     nSumCourseUnit NUMBER;
13     nCourseUnit NUMBER;
14     nCnt NUMBER;
15     nTeachMax NUMBER;
16 BEGIN
17     result := '';
18     DBMS_OUTPUT.put_line('#');
19     DBMS_OUTPUT.put_line(sStudentId || '님이 과목번호 ' || sCourseId ||
20     ', 분반 ' || TO_CHAR(nCourseIdNo) || '의 수강 등록을 요청하였습니다.');
```

```
22  /* 년도, 학기 알아내기 */
23      nYear := Date2EnrollYear(SYSDATE);
24      nSemester := Date2EnrollSemester(SYSDATE);
25
26  /* 예러 처리 1 : 최대학점 초과여부 */
27      SELECT SUM(c.c_unit)
28      INTO      nSumCourseUnit
29      FROM      course c, enroll e
30      WHERE     e.s_id = sStudentId and e.e_year = nYear and e.e_semester = nSemester
31      and       e.c_id = c.c_id and e.c_id_no = c.c_id_no;
32
33      SELECT c_unit
34      INTO      nCourseUnit
35      FROM      course
36      WHERE     c_id = sCourseId and c_id_no = nCourseIdNo;
37
38      IF (nSumCourseUnit + nCourseUnit > 18) THEN
39          RAISE too_many_sumCourseUnit;
40      END IF;
```

함수 사용

사용자 정의 예외

```
42  /* 에러 처리 2 : 동일한 과목 신청 여부 */
43      SELECT COUNT(*)
44      INTO      nCnt
45      FROM      enroll
46      WHERE     s_id = sStudentId and c_id = sCourseId;
47
48      IF (nCnt > 0) THEN
49          RAISE too_many_courses;
50      END IF;
51
52  /* 에러 처리 3 : 수강신청 인원 초과 여부 */
53      SELECT t_max
54      INTO      nTeachMax
55      FROM      teach
56      WHERE     t_year= nYear and t_semester = nSemester and c_id = sCourseId
57      and c_id_no= nCourseIdNo;
58
59      SELECT COUNT(*)
60      INTO      nCnt
61      FROM      enroll
62      WHERE     e_year = nYear and e_semester = nSemester
63      and c_id = sCourseId and c_id_no = nCourseIdNo;
64      IF (nCnt >= nTeachMax) THEN
65          RAISE too_many_students;
66      END IF;
67
```

```
68  /* 예러 처리 4 : 신청한 과목들 시간 중복 여부 */
69      SELECT COUNT(*)
70      INTO    nCnt
71      FROM (SELECT t_time
72            FROM teach
73            WHERE t_year=nYear and t_semester = nSemester
74 and c_id = sCourseId and c_id_no = nCourseIdNo
75            INTERSECT
76            SELECT t.t_time
77            FROM teach t, enroll e
78            WHERE e.s_id=sStudentId and e.e_year=nYear
79            and e.e_semester = nSemester and t.t_year=nYear
80            and t.t_semester = nSemester and e.c_id=t.c_id
81            and e.c_id_no=t.c_id_no);
82      IF (nCnt > 0) THEN
83          RAISE duplicate_time;
84      END IF;
85
```

```
86  /* 수강 신청 등록 */
87  INSERT INTO enroll(S_ID,C_ID,C_ID_NO,E_YEAR,E_SEMESTER)
88  VALUES (sStudentId, sCourseId, nCourseIdNo, nYear, nSemester);
89  COMMIT;
90  result := '수강신청 등록이 완료되었습니다.';
91
92  EXCEPTION
93  WHEN too many sumCourseIdNo THEN
94  result := '최대학점을 초과하였습니다';
95  WHEN too many courses THEN
96  result := '이미 등록된 과목을 신청하였습니다';
97  WHEN too many students THEN
98  result := '수강신청 인원이 초과되어 등록이 불가능합니다';
99  WHEN duplicate_time THEN
100 result := '이미 등록된 과목 중 중복되는 시간이 존재합니다';
101 WHEN OTHERS THEN
102 ROLLBACK;
103 result := SQLCODE;
104 END;
105 /
```

```
SQL> CREATE OR REPLACE PROCEDURE SelectTimeTable
  2      (sStudentId IN VARCHAR2,
  3      nYear IN NUMBER,
  4      nSemester IN NUMBER)
  5  IS
  6      sId COURSE.C_ID%TYPE;
  7      sName COURSE.C_NAME%TYPE;
  8      nIdNo COURSE.C_ID_NO%TYPE;
  9      nUnit COURSE.C_UNIT%TYPE;
 10      nTime TEACH.T_TIME%TYPE;
 11      sWhere TEACH.T_WHERE%TYPE;
 12      nTotUnit NUMBER := 0;
 13  CURSOR cur (sStudentId VARCHAR2, nYear NUMBER, nSemester NUMBER) IS
 14      SELECT e.c_id, c.c_name, e.c_id_no, c.c_unit, t.t_time, t.t_where
 15      FROM enroll e, course c, teach t
 16      WHERE e.s_id = sStudentId and e.e_year = nYear
 17      and e.e_semester=nSemester
 18      and t.t_year = nYear and t.t_semester = nSemester
 19      and e.c_id = c.c_id and e.c_id_no=c.c_id_no
 20      and c.c_id=t.c_id and c.c_id_no = t.c_id_no
 21      ORDER BY 5;
```

명시적 커서 사용



```
22 BEGIN
23   OPEN cur(sStudentId, nYear, nSemester); 파라미터가 있는 커서 사용
24   DBMS_OUTPUT.put_line('#');
25   DBMS_OUTPUT.put_line(TO_CHAR(nYear) || '년도 ' || TO_CHAR(nSemester) ||
26     '학기의 ' || sStudentId || '님의 수강신청 시간표입니다. ');
27   LOOP
28     FETCH cur INTO sId, sName, nIdNo, nUnit, nTime, sWhere;
29     EXIT WHEN cur%NOTFOUND;
30     DBMS_OUTPUT.put_line('교시:' || TO_CHAR(nTime) || ', 과목번호:' || sId ||
31       ', 과목명:' || sName || ', 분반:' || TO_CHAR(nIdNo) ||
32       ', 학점:' || TO_CHAR(nUnit) || ', 장소:' || sWhere);
33     nTotUnit := nTotUnit + nUnit;
34   END LOOP;
35   DBMS_OUTPUT.put_line('총 ' || TO_CHAR(cur%ROWCOUNT) || ' 과목과 총 '
36     || TO_CHAR(nTotUnit) || ' 학점을 신청하였습니다. ');
37   CLOSE cur;
38 END;
39 /
```

프로시저가 생성되었습니다.

- InsertEnroll() 프로시저의 입력 결과 확인
  - InsertEnroll() 프로시저 호출
    - 과목번호 'C400' 입력 : 동일한 과목 신청 오류 발생
    - 과목번호 'C900' 입력 : 수강신청 인원 초과 오류 발생
    - 과목번호 'M100' 입력 : 신청한 과목들 시간 중복 발생
    - 과목번호 'C800' 입력 : 정상적 입력
    - 과목번호 'M700' 입력 : 최대 학점 초과 오류 발생
  - 입력 후 결과 확인
    - SelectTimeTable() 프로시저 호출

```
SQL> DECLARE
2  result VARCHAR2(100) := '';
3  BEGIN
4  DBMS_OUTPUT.enable;
5  DBMS_OUTPUT.put_line('***** Insert 및 에러 처리 테스트 *****');
6
7  /* 에러 처리 2 : 동일한 과목 신청 여부 : 데이터베이스 과목 신청 */
8  InsertEnroll('20011234', 'C400', 3, result);
9  DBMS_OUTPUT.put_line('결과 : ' || result);
10
11 /* 에러 처리 3 : 수강신청 인원 초과 여부 : 객체지향 윈도우즈 프로그래밍 신청*/
12 InsertEnroll('20011234', 'C900', 3, result);
13 DBMS_OUTPUT.put_line('결과 : ' || result);
14
15 /* 에러 처리 4 : 신청한 과목들 시간 중복 여부 : 멀티미디어 개론 신청 */
16 InsertEnroll('20011234', 'M100', 3, result);
17 DBMS_OUTPUT.put_line('결과 : ' || result);
18
19 /* 에러가 없는 경우 */
20 InsertEnroll('20011234', 'C800', 3, result);
21 DBMS_OUTPUT.put_line('결과 : ' || result);
22
23 /* 에러 처리 1 : 최대 학점 초과 여부 검사 : 게임 프로그래밍 신청 */
24 InsertEnroll('20011234', 'M700', 3, result);
25 DBMS_OUTPUT.put_line('결과 : ' || result);
26 DBMS_OUTPUT.put_line('***** CURSOR를 이용한 SELECT 테스트 *****');
27
28 /* 최종 결과 확인 */
29 SelectTimeTable('20011234', 2023, 1);
30 delete from enroll where s_id='20011234' and c_id='C800' and c_id_no=3;
31 END;
32 /
```

\*\*\*\*\* Insert 및 에러 처리 테스트 \*\*\*\*\*

#  
20011234님이 과목번호 C400, 분반 3의 수강 등록을 요청하였습니다.  
결과 : 이미 등록된 과목을 신청하였습니다

#  
20011234님이 과목번호 C900, 분반 3의 수강 등록을 요청하였습니다.  
결과 : 수강신청 인원이 초과되어 등록이 불가능합니다

#  
20011234님이 과목번호 M100, 분반 3의 수강 등록을 요청하였습니다.  
결과 : 이미 등록된 과목 중 중복되는 시간이 존재합니다

#  
20011234님이 과목번호 C800, 분반 3의 수강 등록을 요청하였습니다.  
결과 : 수강신청 등록이 완료되었습니다.

#  
20011234님이 과목번호 M700, 분반 3의 수강 등록을 요청하였습니다.  
결과 : 최대학점을 초과하였습니다

\*\*\*\*\* CURSOR를 이용한 SELECT 테스트 \*\*\*\*\*

#  
2023년도 1학기의 20011234님의 수강신청 시간표입니다.  
교시:2, 과목번호:C300, 과목명:알고리즘, 분반:3, 학점:3, 장소:인-416  
교시:3, 과목번호:C500, 과목명:운영체제, 분반:3, 학점:3, 장소:인-201  
교시:4, 과목번호:C100, 과목명:컴퓨터 프로그래밍, 분반:3, 학점:3, 장소:인-201  
교시:5, 과목번호:C200, 과목명:자료구조, 분반:3, 학점:3, 장소:인-201  
교시:6, 과목번호:C400, 과목명:데이터베이스 시스템, 분반:3, 학점:3, 장소:인-201  
교시:7, 과목번호:C800, 과목명:데이터베이스 프로그래밍, 분반:3, 학점:3, 장소:인-309  
총 6 과목과 총 18학점을 신청하였습니다.

PL/SQL 처리가 정상적으로 완료되었습니다.

```
SQL> CREATE OR REPLACE TRIGGER BeforeUpdateStudent
  2 BEFORE UPDATE ON student
  3 FOR EACH ROW
  4 DECLARE
  5     underflow_length EXCEPTION;
  6     invalid_value EXCEPTION;
  7     nLength NUMBER;
  8     nBlank NUMBER;
  9 BEGIN
10     /* 학년 제약조건 : DDL에서 해결 */
11     /* 보다 복잡한 제약조건을 다루기 편하게 함 */
12     /* 암호 제약조건 : 4자리 이상, blank는 허용안함 */
13     SELECT length(:new.s_pwd), instr(:new.s_pwd, ' ')
14     INTO nLength, nBlank
15     FROM DUAL;
16     IF (nLength < 4) THEN
17         RAISE underflow_length;
18     ELSIF (nBlank > 0) THEN
19         RAISE invalid_value;
20     END IF;
21 EXCEPTION
22     WHEN underflow_length THEN
23         RAISE_APPLICATION_ERROR(-20002, '암호는 4자리 이상이어야 합니다');
24     WHEN invalid_value THEN
25         RAISE_APPLICATION_ERROR(-20003, '암호에 공란은 입력되지 않습니다. ');
26 END;
27 /
```

트리거가 생성되었습니다.

```
SQL> UPDATE student set s_pwd = '12' where s_id='20011234';  
UPDATE student set s_pwd = '12' where s_id='20011234'
```

\*

1행에 오류:

ORA-20002: 암호는 4자리 이상이어야 합니다

ORA-06512: "DBBK.BEFOREUPDATESTUDENT", 20행

ORA-04088: 트리거 'DBBK.BEFOREUPDATESTUDENT'의 수행시 오류

```
SQL> UPDATE student set s_pwd = '13 45' where s_id='20011234';  
UPDATE student set s_pwd = '13 45' where s_id='20011234'
```

\*

1행에 오류:

ORA-20003: 암호에 공란은 입력되지 않습니다.

ORA-06512: "DBBK.BEFOREUPDATESTUDENT", 23행

ORA-04088: 트리거 'DBBK.BEFOREUPDATESTUDENT'의 수행시 오류

- 설치 프로그램

- Apache Tomcat8

- 버전 : 8.5.83

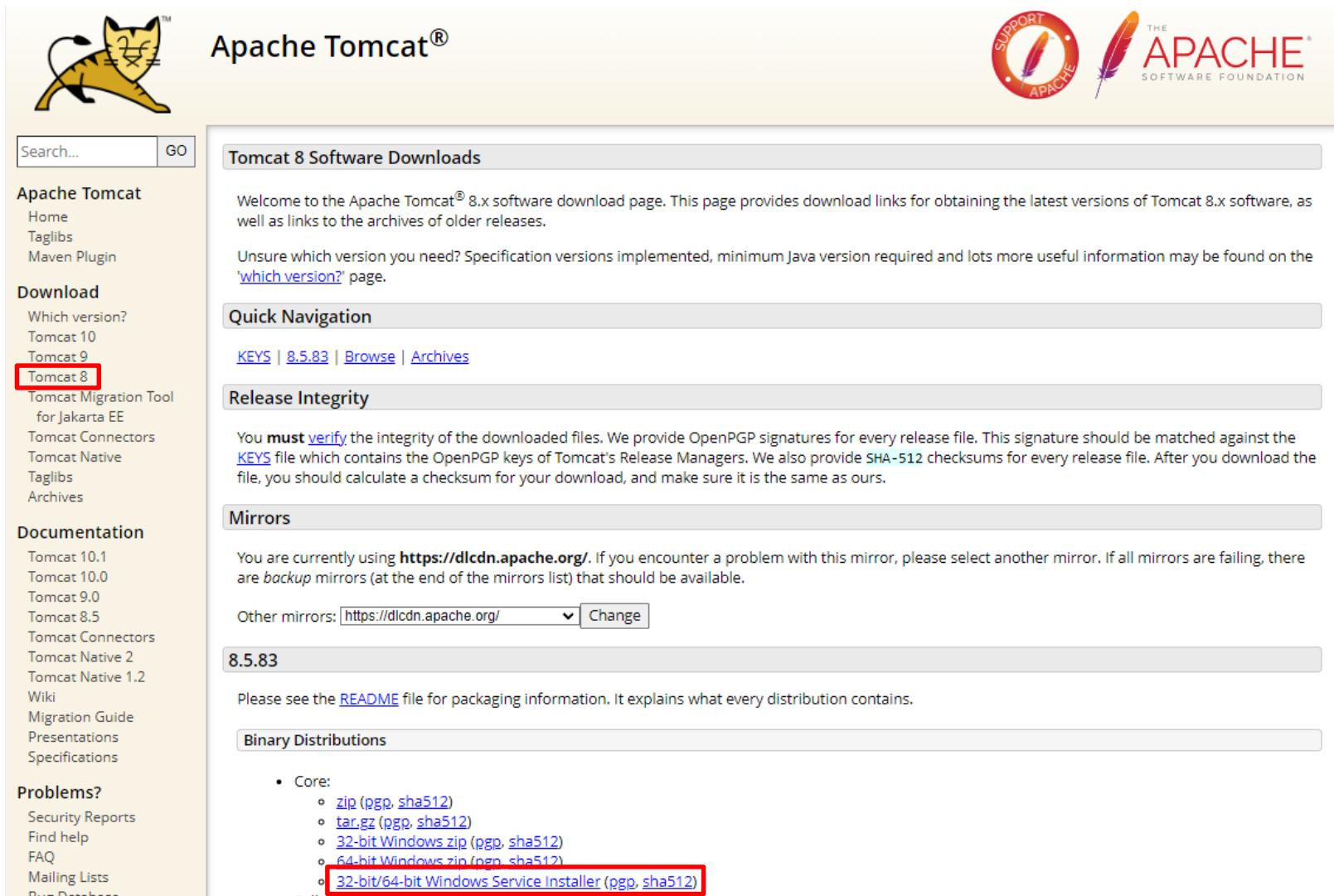
- Java8 SE Development Kit

- 버전 : 1.8.0\_341

- ※ 주의사항

- Java11부터 정책 변경으로 인해 JRE를 사용할 수 없어 JDBC 드라이버를 **Apache Software Foundation\Tomcat\webapps\ROOT\WEB-INF\lib** 폴더에 넣어야 함.
    - lib폴더가 없을 경우, 별도로 폴더를 생성해야 함.

- <http://tomcat.apache.org>



The screenshot shows the Apache Tomcat website's download page for version 8. The page has a yellow header with the Tomcat logo and the Apache Software Foundation logo. A search bar is located in the top left. The left sidebar contains a navigation menu with links to Home, Taglibs, Maven Plugin, Download, Documentation, and Problems?. The 'Download' section is expanded, showing links for 'Which version?', 'Tomcat 10', 'Tomcat 9', and 'Tomcat 8', with 'Tomcat 8' highlighted by a red box. The main content area is titled 'Tomcat 8 Software Downloads' and contains a welcome message, a 'Quick Navigation' section with links to KEYS, 8.5.83, Browse, and Archives, a 'Release Integrity' section explaining how to verify downloads, a 'Mirrors' section with a dropdown menu showing 'https://dlcdn.apache.org/' and a 'Change' button, and a 'Binary Distributions' section listing various download options. The '32-bit/64-bit Windows Service Installer (pgp, sha512)' link is highlighted by a red box.

Apache Tomcat®

Search... GO

Apache Tomcat

- Home
- Taglibs
- Maven Plugin

Download

- Which version?
- Tomcat 10
- Tomcat 9
- Tomcat 8**
- Tomcat Migration Tool for Jakarta EE
- Tomcat Connectors
- Tomcat Native
- Taglibs
- Archives

Documentation

- Tomcat 10.1
- Tomcat 10.0
- Tomcat 9.0
- Tomcat 8.5
- Tomcat Connectors
- Tomcat Native 2
- Tomcat Native 1.2
- Wiki
- Migration Guide
- Presentations
- Specifications

Problems?

- Security Reports
- Find help
- FAQ
- Mailing Lists
- Bug Database

## Tomcat 8 Software Downloads

Welcome to the Apache Tomcat® 8.x software download page. This page provides download links for obtaining the latest versions of Tomcat 8.x software, as well as links to the archives of older releases.

Unsure which version you need? Specification versions implemented, minimum Java version required and lots more useful information may be found on the ["which version?"](#) page.

### Quick Navigation

[KEYS](#) | [8.5.83](#) | [Browse](#) | [Archives](#)

### Release Integrity

You **must** [verify](#) the integrity of the downloaded files. We provide OpenPGP signatures for every release file. This signature should be matched against the [KEYS](#) file which contains the OpenPGP keys of Tomcat's Release Managers. We also provide [SHA-512](#) checksums for every release file. After you download the file, you should calculate a checksum for your download, and make sure it is the same as ours.

### Mirrors

You are currently using <https://dlcdn.apache.org/>. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, there are [backup mirrors](#) (at the end of the mirrors list) that should be available.

Other mirrors:

## 8.5.83

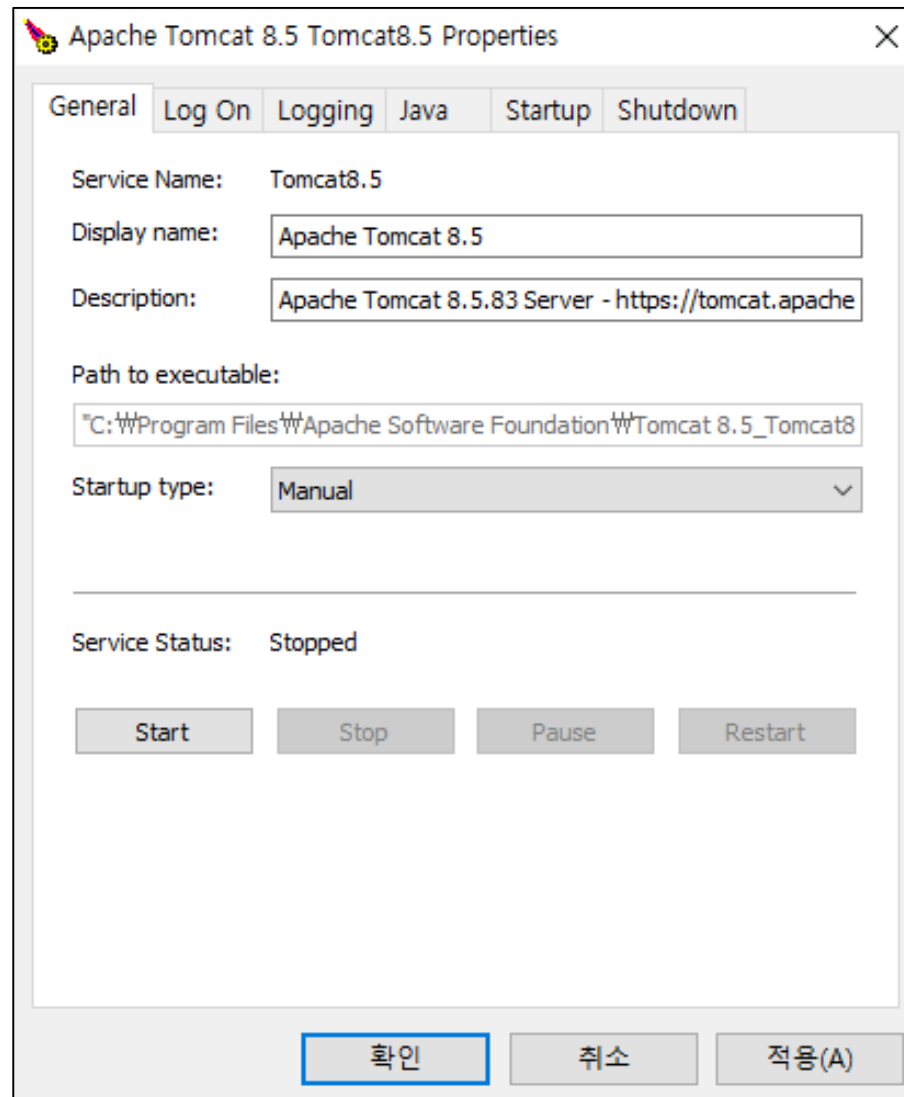
Please see the [README](#) file for packaging information. It explains what every distribution contains.

### Binary Distributions

- Core:
  - [zip \(pgp, sha512\)](#)
  - [tar.gz \(pgp, sha512\)](#)
  - [32-bit Windows zip \(pgp, sha512\)](#)
  - [64-bit Windows zip \(pgp, sha512\)](#)
  - [32-bit/64-bit Windows Service Installer \(pgp, sha512\)](#)**




- Service Start | Stop




- <http://localhost:8080/>  
문제없이 설치 및 아파치가 실행되었다면 아래와 같은 페이지가 나옴.

[Home](#) [Documentation](#) [Configuration](#) [Examples](#) [Wiki](#) [Mailing Lists](#) [Find Help](#)

**Apache Tomcat/8.5.83** 

If you're seeing this, you've successfully installed Tomcat. Congratulations!



**Recommended Reading:**  
[Security Considerations How-To](#)  
[Manager Application How-To](#)  
[Clustering/Session Replication How-To](#)

[Server Status](#)  
[Manager App](#)  
[Host Manager](#)

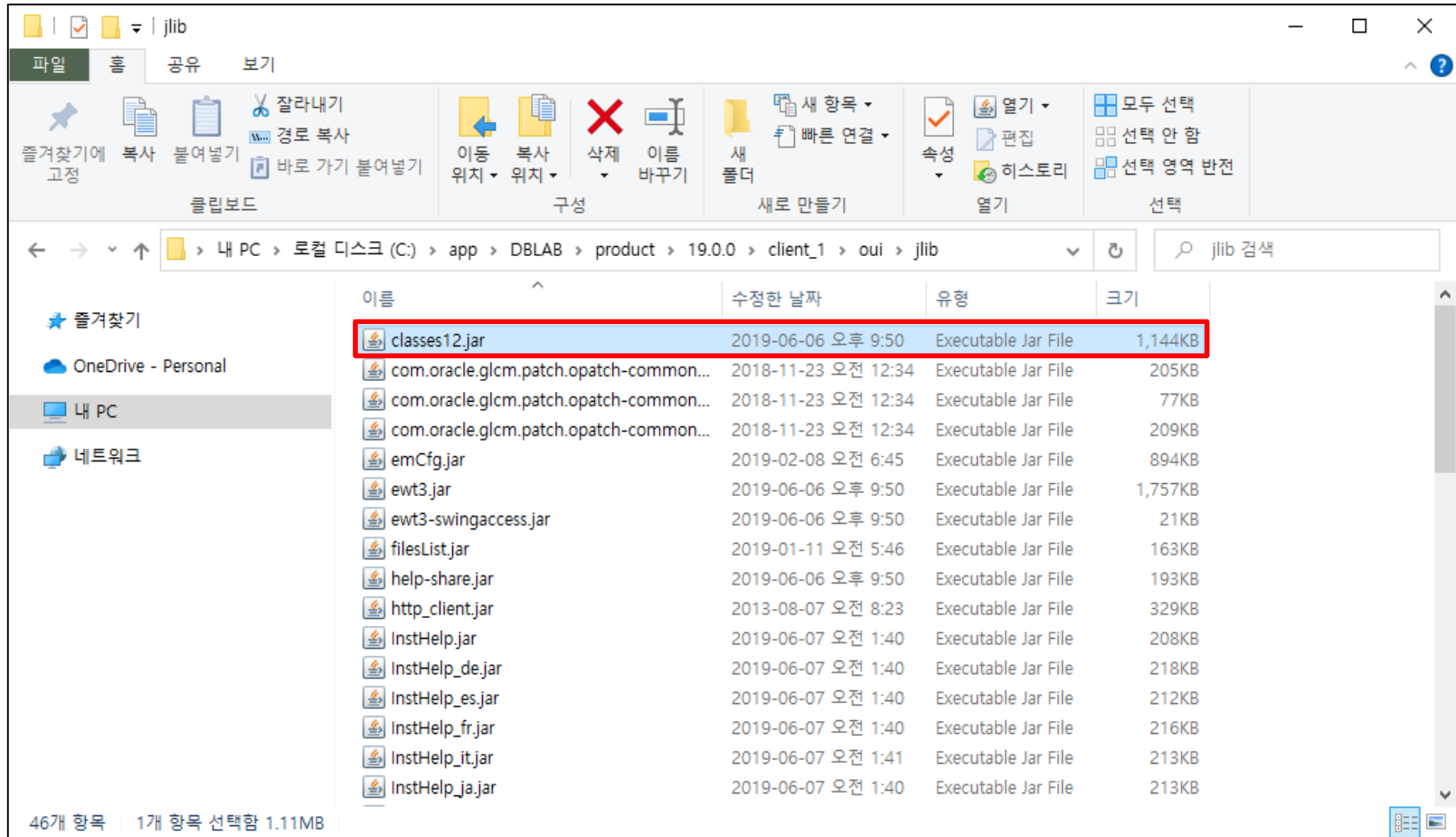
**Developer Quick Start**  
[Tomcat Setup](#) [Realms & AAA](#) [Examples](#) [Servlet Specifications](#)  
[First Web Application](#) [JDBC DataSources](#) [Tomcat Versions](#)

**Managing Tomcat**  
For security, access to the [manager webapp](#) is restricted. Users are defined in:  
`$CATALINA_HOME/conf/tomcat-users.xml`  
In Tomcat 8.5 access to the manager application is split between different users.  
[Read more...](#)  
[Release Notes](#)  
[Changelog](#)  
[Migration Guide](#)  
[Security Notices](#)

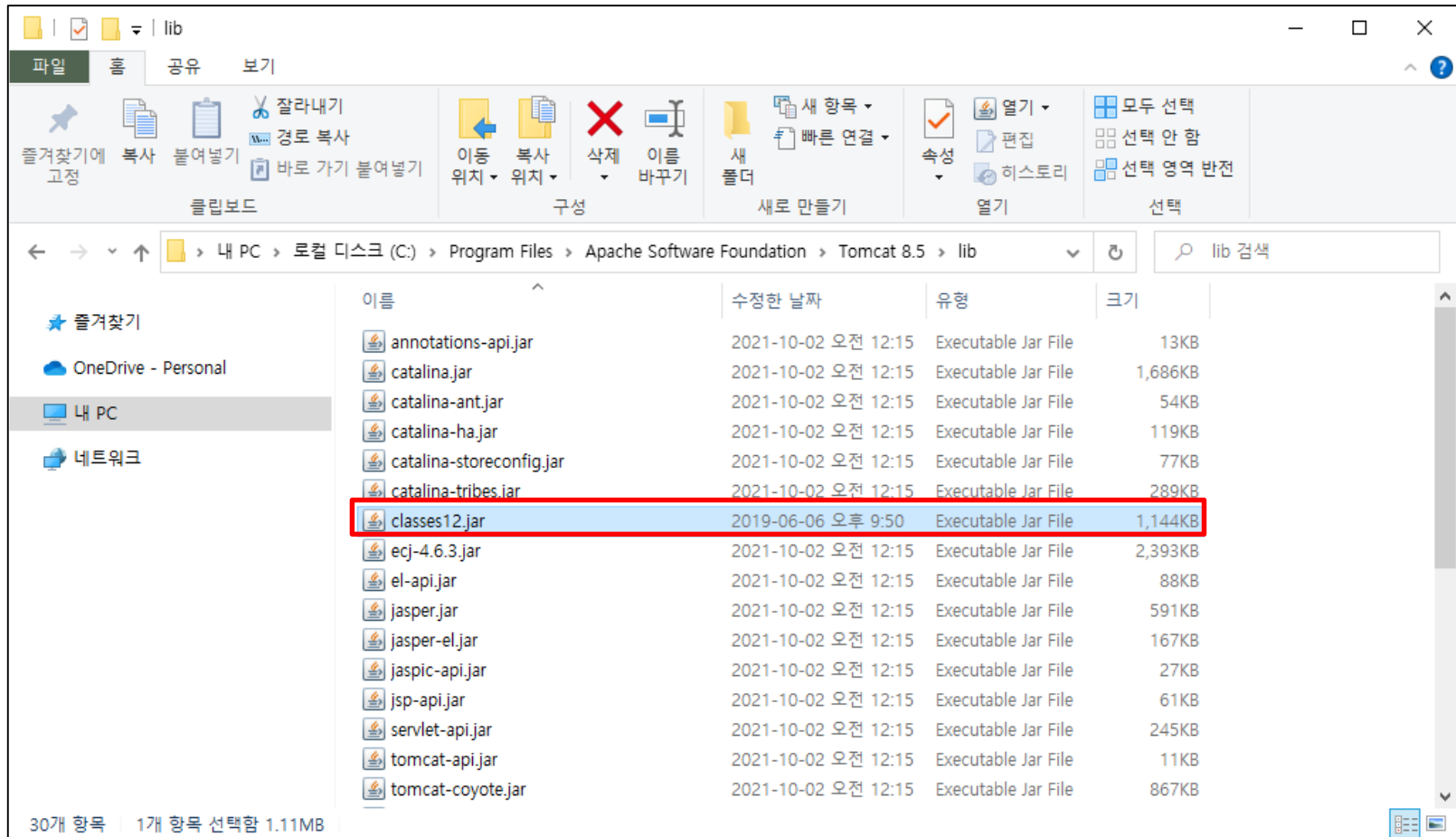
**Documentation**  
[Tomcat 8.5 Documentation](#)  
[Tomcat 8.5 Configuration](#)  
[Tomcat Wiki](#)  
Find additional important configuration information in:  
`$CATALINA_HOME/RUNNING.txt`  
Developers may be interested in:  
[Tomcat 8.5 Bug Database](#)  
[Tomcat 8.5 JavaDocs](#)  
[Tomcat 8.5 Git Repository at GitHub](#)

**Getting Help**  
**FAQ and Mailing Lists**  
The following mailing lists are available:  
[tomcat-announce](#)  
Important announcements, releases, security vulnerability notifications. (Low volume).  
[tomcat-users](#)  
User support and discussion  
[taglibs-user](#)  
User support and discussion for [Apache Taglibs](#)  
[tomcat-dev](#)  
Development mailing list, including commit messages

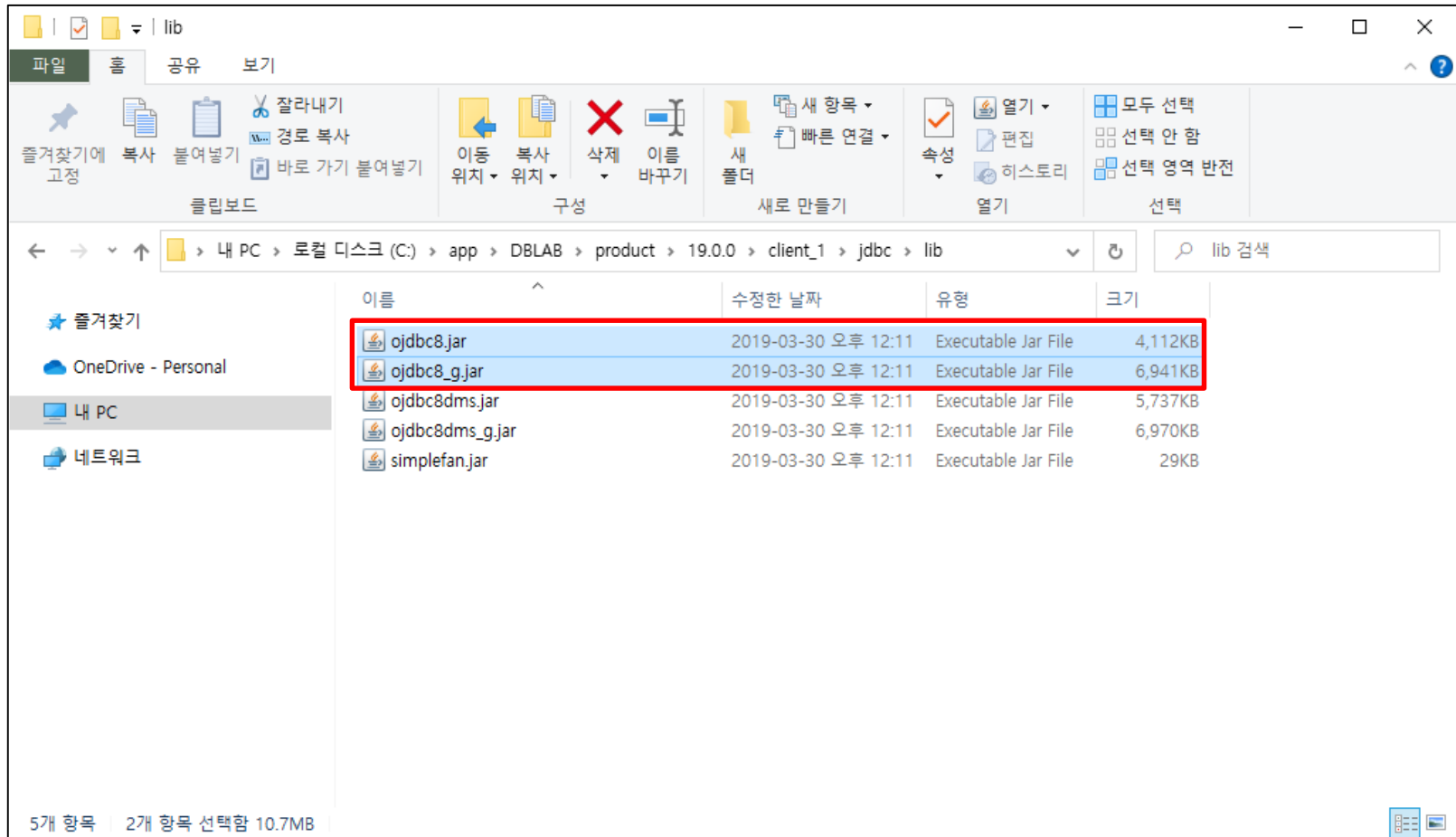
- C:\app\user\product\19.0.0\client\_1\oui\jlib



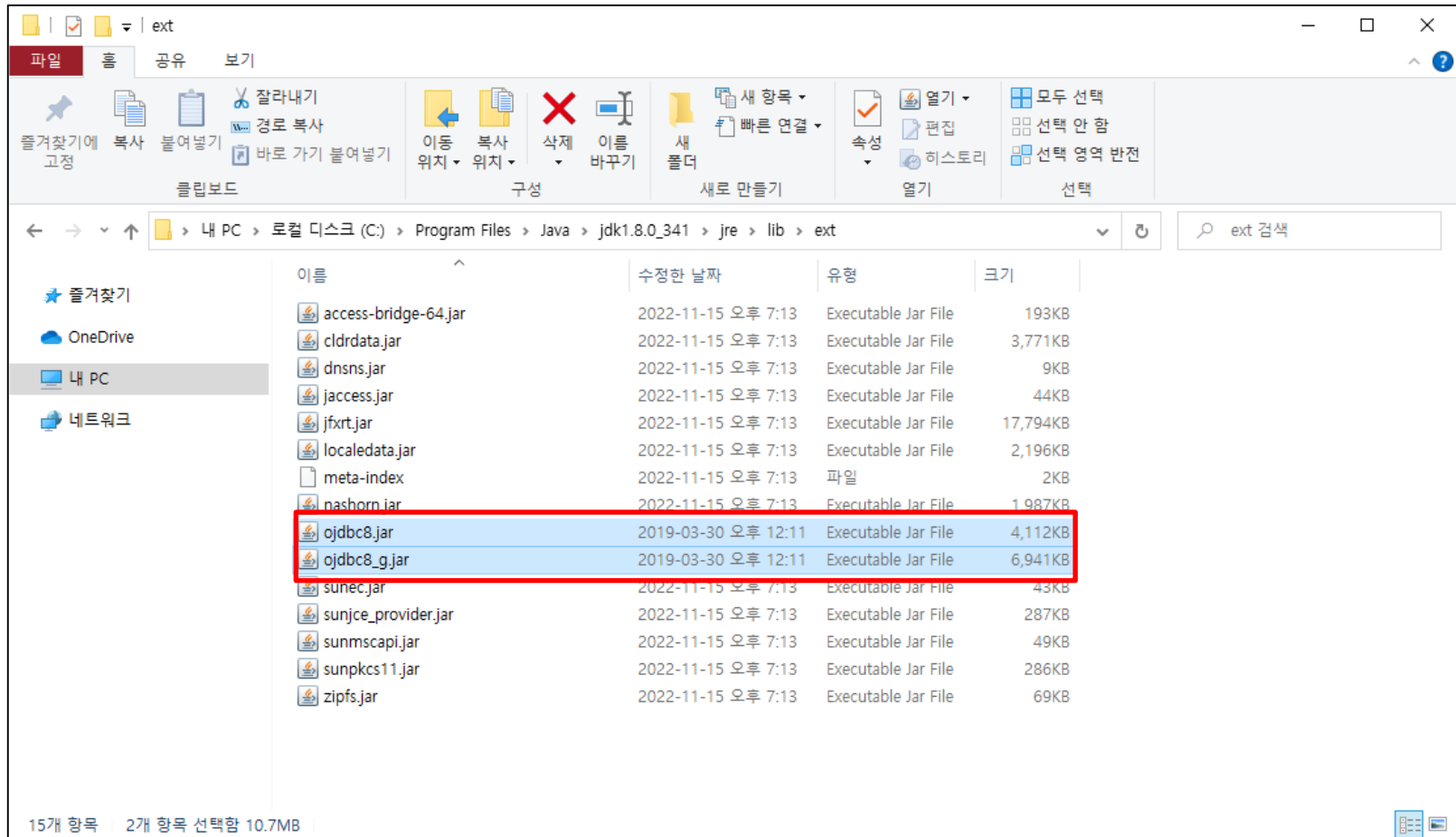
- C:\Program Files\Apache Software Foundation\Tomcat 8.5\lib로 classes12.jar 파일 복사 이동

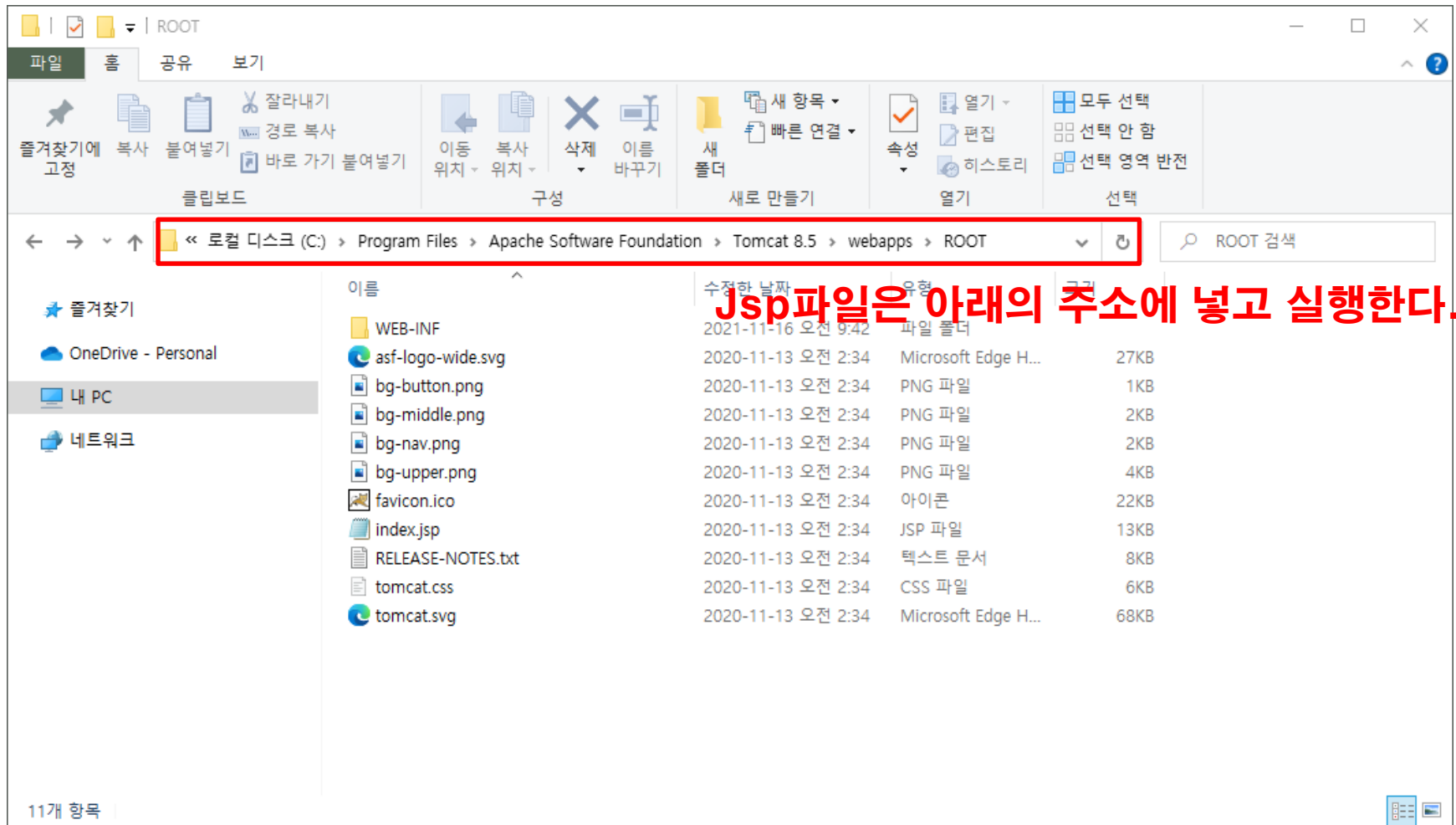


- C:\app\user\product\19.0.0\client\_1\jdbc\lib



- C:\Program Files\Java\jdk1.8.0\_341\jre\lib\ext로 ojdbc8.jar, ojdbc8\_g.jar 파일 2개 복사 이동





## ■ JDBC 드라이버를 이용한 데이터베이스 연결

- Connection con = DriverManager.getConnection  
("jdbc:oracle:thin:@ServerName:1521:SID","사용자 계정","비밀번호");
  - ServerName : 210.94.199.20
  - SID : dblab
  - 사용자 계정 : ST학번
  - 비밀번호 : ST학번



## • 연결 확인 예시 (connect.jsp)

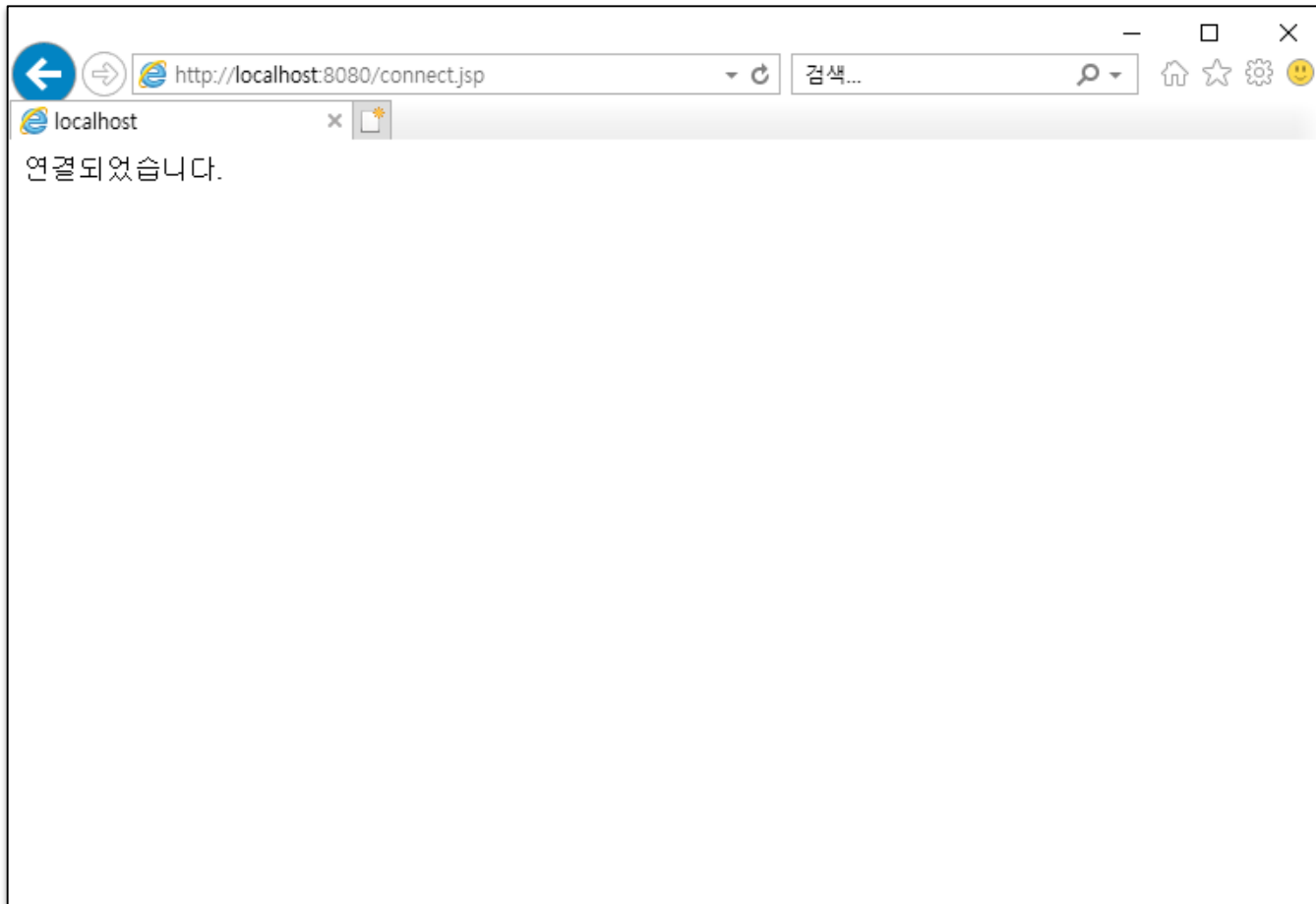
```
<%@ page language="java" contentType="text/html; charset=utf-8" pageEncoding="utf-8" %>
<%@ page import="java.sql.*" %>
<%
    Connection conn = null;

    String driver = "oracle.jdbc.driver.OracleDriver";
    String url = "jdbc:oracle:thin:@210.94.199.20:1521:dblab";

    Boolean connect = false;

    try{
        Class.forName(driver);
        conn = DriverManager.getConnection(url,"id","password");
        connect = true;
        conn.close();
    }
    catch(Exception e){
        connect= false;
        out.print(e);
    }
%>
<html>
<head></head>
<body>
<% if(connect == true) {
    out.print("연결되었습니다.");
} else {
    out.print("연결에 실패하였습니다.");
}%>
</body>
</html>
```

- 연결 확인 예시



< 연결 성공 화면 >

**Q1>** 아래 'InsertEnroll' 프로시저의 일부분을 명시적 커서로 변경하시오.

```
/* 예러 처리 3 : 수강신청 인원 초과 여부 */  
SELECT t_max  
  INTO  nTeachMax  
FROM    teach  
WHERE   t_year= nYear and t_semester = nSemester and c_id = sCourseId  
and c_id_no= nCourseIdNo;  
  
SELECT COUNT(*)  
  INTO  nCnt  
FROM    enroll  
WHERE   e_year = nYear and e_semester = nSemester  
and c_id = sCourseId and c_id_no = nCourseIdNo;  
IF (nCnt >= nTeachMax) THEN  
    RAISE too_many_students;  
END IF;
```

**Q2>** 사원번호를 입력 받아 사원의 이름을 출력하고 EMP 테이블에서 해당 사원의 정보를 삭제하는 프로시저를 작성하고 실행하시오. 단, 삭제하고자 하는 사원이 관리하는 부하 직원이 있는 경우, 해당 부하직원의 관리자(MGR)를 NULL로 변경하고, 변경한 부하 직원 정보를 출력한다. 또한 부하직원이 없는 경우, 사용자 정의 예외로 '부하직원이 없는 사원입니다.' 라는 메시지를 출력하도록 한다. 추가로 존재하지 않는 사원번호를 입력할 경우, 미리 정의된 예외로 '존재하지 않는 사원입니다.' 라는 메시지를 출력한다.

(%TYPE, 파라미터가 있는 커서, LOOP, IF문 사용)

※ 프로시저 실행 후 emp 테이블을 조회하시오.

```
SQL> EXECUTE emp_delete(7566);
삭제할 사원 이름 : JONES
```

<부하직원 정보>

사원번호	이름	부서명	직책	관리자번호
7788	SCOTT	RESEARCH	ANALYST	7566
7902	FORD	RESEARCH	ANALYST	7566

줄바꿈 :chr(10)||chr(13)

데이터 공간 맞춤 : rpad함수 사용  
Ex) rpad(컬럼명, 데이터 길이(숫자))

```
SQL> execute emp_delete(7934);
```

삭제할 사원 이름 : MILLER

부하직원이 없는 사원입니다.

```
SQL> execute emp_delete(9999);
```

존재하지 않는 사원입니다.

- 제출 방식 : E-Class를 통하여 제출
- 제출 내용 : spool file
- 제출 형식 : 학번\_이름\_주차
  - Ex) 학번\_홍길동\_11주차.sql
- 제출 기한 : 수업 시작 시간으로 부터 24시간 이내 제출
  - 제출 기한 위반 시 감점 기준
    - 지각 제출 시 과제 점수에서 40% 감점
    - 1일 초과 당 20% 추가 감점 (단, 4일 이후 제출 불가)