

05. 데이터베이스

05.데이터베이스

학습 목표

데이터베이스 관리 시스템(DBMS) 중에서 오라클을 설치하고 JDBC API를 이용해서 JSP와 연동하는 방법을 학습합니다.

학습 순서

- 데이터베이스 소개
- 오라클 설치 및 계정 생성
- MySQL 설치 및 계정 생성
- JDBC를 이용한 DB연결
- 커넥션풀을 이용한 DB연결
- JSP에서 쿼리작성 및 실행

데이터베이스란?

일반적으로 컴퓨터 시스템에 전자적으로 저장되는 구조화된 정보 또는 데이터의 조직화된 모음




- 우리가 매일 PC나 스마트폰을 통해 접하는 거의 모든 웹 애플리케이션은 데이터베이스를 사용
- 정보를 제공하는 측에서는 필요한 내용을 데이터베이스에 미리 입력
- 클라이언트는 콘텐츠를 웹 페이지를 통해 확인
- JSP에서는 JDBC(Java Database Connectivity)를 통해 데이터베이스와 연동

오라클 설치

오라클 홈페이지 접속

. <https://www.oracle.com/database/technologies/xe-prior-release-downloads.html>

윈도우 64비트용 다운로드 및 라이선스 동의

Oracle Database XE Prior Release Archive	
Oracle Database Express Edition (XE) Release 11.2.0.2.0 (11gR2).	
Name	Download
Oracle Database 11gR2 Express Edition for Windows x64	 Download
Oracle Database 11gR2 Express Edition for Windows x32	 Download
Oracle Database 11gR2 Express Edition for Linux x64	 Download

✕

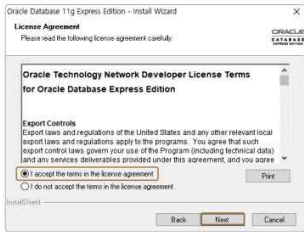
You must accept the Oracle License Agreement to download this software.

☒ I reviewed and accept the Oracle License Agreement

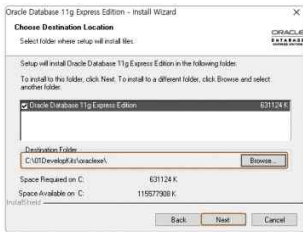
You will be redirected to the login screen in order to download the file.

Download OracleXE112_Win64.zip 

오라클 설치

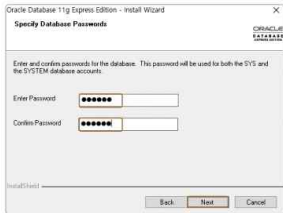


- . 압축해제 후 setup.exe 더블클릭
- . 라이선스 동의

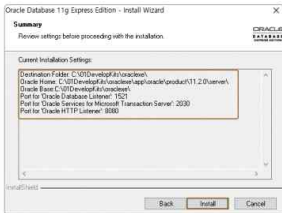


- . C:\01DevelopKits 하위에 oraclexe 폴더 생성
- . 개발 관련된 도구를 한 곳에서 관리하기 위함

오라클 설치



- 관리계정 system의 패스워드 입력
- 학습용이므로 123456로 설정



- 지금까지의 설정 요약
- [Install]을 클릭하면 설치가 진행

사용자 계정 생성 및 권한 설정



- 명령 프롬프트 실행
- sqlplus 명령 실행
- user-name : system
- password : 123456 입력 후 접속

```
SQL> create user musthave identified by 1234;  
User created.
```

- 실습용 계정 생성
- musthave / 1234

```
SQL> create user musthave identified by 1234;  
User created.
```

- 롤(Role)을 통한 권한 부여
- 접속과 객체생성에 필요한 권한을 묶어놓은것

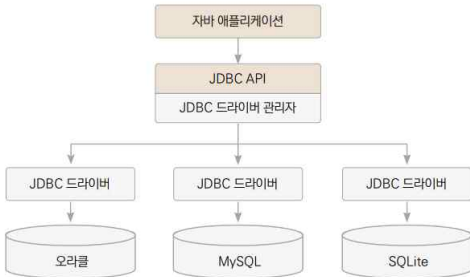
MySQL 설치

- <https://www.mysql.com/>에서
- MySQL Community (GPL) Downloads » 다운로드하여 설치
- 사용자 계정 및 권한 설정

JDBC 설정 및 데이터베이스 연결

JDBC(Java Database Connectivity)

- 자바로 데이터베이스 연결 및 관련 작업을 할 때 사용하는 API
- JDBC API는 JDBC 드라이버가 있어야 하므로 다운로드 및 설정 후 사용
- 우리는 오라클(Oracle XE) 사용



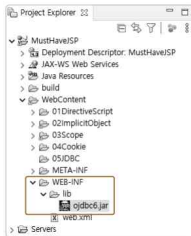
JDBC 설정 및 데이터베이스 연결

■ JDBC 드라이버 설정

- 우리는 오라클을 이미 설치하였으므로 드라이버 파일은 별도의 다운로드 필요없음
- 경로
C:\01DevelopKits\oraclexe\app\oracle\product\11.2.0\server\jdbc\lib



- 설치방법
 - 드라이버 파일을 프로젝트의 WEB-INF 하위의 lib 폴더에 추가
 - 단순히 복사하여 추가하는것 만으로 모든 설정 완료
 - 배포 시에도 드라이버는 함께 따라가므로 별도의 설정 필요없음



JDBC 설정 및 데이터베이스 연결

■ 연결 관리 클래스 작성

```
public class JDBCConnect {  
    Connection ← DB연결  
    Statement, PreparedStatement, ResultSet ← 쿼리실행 및 결과  
    public JDBCConnect() {  
        try { ← 오라클 드라이버 및 커넥션URL, 계정 정보  
            Class.forName("oracle.jdbc.OracleDriver");  
            String url = "jdbc:oracle:thin:@localhost:1521:xe";  
            String id = "musthave";  
            String pwd = "1234";  
            con = DriverManager.getConnection(url, id, pwd);  
        }  
        catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

com.mysql.jdbc.Driver

"jdbc:mysql://localhost/dev?useSSL=false
&serverTimezone=Asia/Seoul&characterE
ncoding=UTF-8"

▶ 커넥션URL 구성 형식



예제 5-6 : JDBCConnect.java

JDBC 설정 및 데이터베이스 연결

■ 연결 설정 개선1

```
<context-param>
  <param-name>OracleDriver</param-name> ← 드라이버 이름
  <param-value>oracle.jdbc.OracleDriver</param-value>
</context-param>
<context-param>
  <param-name>OracleURL</param-name> ← 접속 URL
  <param-value>jdbc:oracle:thin:@localhost:1521:xe</param-value>
</context-param>
<context-param>
  <param-name>OracleId</param-name> ← 계정 아이디
  <param-value>musthave</param-value>
</context-param>
<context-param>
  <param-name>OraclePwd</param-name> ← 패스워드
  <param-value>1234</param-value>
</context-param>
```

- web.xml에 컨텍스트 초기화 매개변수로 입력
- application 내장객체를 통해 얻어온다.

JDBC 설정 및 데이터베이스 연결

■ 연결 설정 개선1

```
public JDBCConnect(String driver, String url, String id, String pwd) {  
    try { ← 생성자 코드가 이와 같이 간소화  
        Class.forName(driver);  
        con = DriverManager.getConnection(url, id, pwd);  
        System.out.println("DB 연결 성공(인수 생성자 1)");  
    }  
    catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

<h2>JDBC 테스트 2</h2>

<% ← application내장객체로 컨텍스트 초기화 매개변수 얻어옴

String driver = application.getInitParameter("OracleDriver");

String url = application.getInitParameter("OracleURL");

String id = application.getInitParameter("OracleId");

String pwd = application.getInitParameter("OraclePwd");

JDBCConnect jdbc2 = new JDBCConnect(driver, url, id, pwd);

jdbc2.close();

%>

· 접속정보 변경시 web.xml만 수정하면
 됨
· 하지만 JSP에서 반복적인 코드가 발생
 됨

예제5-9 : JDBCConnect.java

예제5-10 : ConnectionTest.jsp

JDBC 설정 및 데이터베이스 연결

■ 연결 설정 개선2

```
public JDBCConnect(ServletContext application) { ← 생성자 수정
    try { ← 생성자에서 직접 web.xml의 내용 읽어옴
        String driver = application.getInitParameter("OracleDriver");
        Class.forName(driver);

        String url = application.getInitParameter("OracleURL");
        String id = application.getInitParameter("OracleId");
        String pwd = application.getInitParameter("OraclePwd");
        con = DriverManager.getConnection(url, id, pwd);
    }
}
```

- web.xml은 그대로 사용
- JSP에서 반복적인 코드를 생성자로 이동
- JSP의 코드 간소화

<h2>JDBC 테스트 3</h2>

<% ← JSP에서 application 내장객체를 인수로 전달

```
JDBCConnect jdbc3 = new JDBCConnect(application);
jdbc3.close();
%>
```

예제5-11 : JDBCConnect.java

예제5-12 : ConnectionTest.jsp

커넥션 풀로 성능 개선

커넥션 풀(connection pool)

- 웹 서버는 클라이언트의 요청시 마다 DB 연결 및 해제를 반복
- 빈번한 연결과 해제는 시스템 성능을 영향을 미침
- 이 문제에 대한 해법으로 커넥션 풀이 널리 사용됨
- Connection 객체를 미리 생성해 풀(pool)에 넣어놓고, 요청이 있을 때 가져다 사용하는 기법

■ JNDI(Java Naming and Directory Interface)

- 자바 소프트웨어에서 객체나 데이터를 전체 경로를 몰라도 '이름'만으로 찾아 쓸 수 있는 디렉터리 서비스



커넥션 풀로 성능 개선

■ JNDI를 통한 커넥션 풀 사용 절차

- WAS(톰캣)가 시작할 때 server.xml과 context.xml에 설정한 대로 커넥션 풀을 생성
- JSP 코드에서 JNDI 서버(WAS가 제공)로부터 데이터소스 객체를 얻어옴
- 데이터소스로부터 커넥션 객체를 가져옴
- DB 작업 수행
- 모든 작업이 끝나면 커넥션 객체를 풀로 반환

■ server.xml 수정

- <GlobalNamingResources> 엘리먼트에 <Resource> 항목 추가
 - driverClassName, type
 - JDBC 드라이버 및 DataSource 클래스 설정
 - minIdle, maxTotal, maxIdle, maxWaitMillis
 - 개체의 개수 및 대기시간 설정
 - url, name, username, password
 - 오라클 연결을 위한 URL, 커넥션풀의 이름, 계정 정보

커넥션 풀로 성능 개선

■ context.xml 수정

- <Context> 엘리먼트에 <ResourceLink> 항목 추가
 - global, name
 - 커넥션풀의 이름 및 전역적으로 사용할 이름 지정
 - type
 - DataSource 클래스 설정

■ JSP 코드와 xml파일간의 관계

JSP 코드

```
context.lookup("자원 이름")
```

context.xml

```
<Context>  
  <ResourceLink  
    global="전역 자원 이름"  
    name="자원 이름"  
    type=".."/>  
</Context>
```

server.xml

```
<GlobalNamingResources>  
  <Resource auth="Container"  
    :  
    name="전역 자원 이름"  
    :  
  />  
</GlobalNamingResources>
```

간단한 쿼리 작성 및 실행

■ Statement 계열의 인터페이스

- . Statement : 인파라미터가 없는 정적 쿼리를 처리할 때 사용
- . PreparedStatement : 인파라미터가 있는 동적 쿼리를 처리할 때 사용
- . CallableStatement : 프로시저(procedure)나 함수(function)를 호출할 때 사용

■ 쿼리문 실행시 사용할 메서드

- . executeUpdate()
 - . INSERT, UPDATE, DELETE 쿼리문을 실행할 때 사용
 - . 기존 레 코드를 변화시키거나 새로운 레코드를 입력하는 쿼리문
 - . 따라서 실행 후 영향을 받은 행의 개수가 int형으로 반환
- . executeQuery()
 - . SELECT 쿼리문을 실행할 때 사용
 - . SELECT는 기존 레코드를 조회하므로, 레코드들의 집합인 ResultSet 객체를 반환

간단한 쿼리 작성 및 실행

동적 쿼리문으로 회원 추가

```
<%  
JDBCConnect jdbc = new JDBCConnect(); ← DB연결  
  
String id, pass, name; ← 테스트용 입력값 준비  
  
String sql = "INSERT INTO member VALUES (?, ?, ?, sysdate)";  
PreparedStatement pstmt = jdbc.con.prepareStatement(sql);  
pstmt.setString(1, id);  
pstmt.setString(2, pass);  
pstmt.setString(3, name); ← 쿼리문 작성 및 인파라미터 설정  
  
int inResult = pstmt.executeUpdate(); ← 쿼리문 실행  
  
jdbc.close(); ← 연결 닫기  
%>
```

간단한 쿼리 작성 및 실행

■ 정적 쿼리문으로 회원 조회

```
<%  
JDBCConnect jdbc = new JDBCConnect(); ← DB연결  
  
String sql = "SELECT id, pass, name, regdate FROM member";  
Statement stmt = jdbc.con.createStatement();  
ResultSet rs = stmt.executeQuery(sql); ← 쿼리 작성 및 실행  
  
while (rs.next()) { ← 결과 반환 및 인출  
    String id = rs.getString(1);  
    String pw = rs.getString(2);  
    String name = rs.getString("name");  
    java.sql.Date regdate = rs.getDate("regdate");  
  
    out.println(인출된 결과를 String으로 묶어 출력);  
}  
jdbc.close(); ← 연결 닫기  
%>
```

간단한 쿼리 작성 및 실행

■ ResultSet에서 결과값 불러오기

- SELECT문의 실행 결과로 얻은 ResultSet에는 일반적으로 여러 개의 레코드가 담겨 있다.
- ResultSet안의 쿼리 결과에는 커서가 첫 번째 행 윗부분에 위치
- next() 메서드를 통해 다음 행으로 커서를 이동
- 이때 다음 행이 있다면 true, 없다면 false를 반환

최초 커서 위치

컬럼 인덱스

	1	2	3	4
	id	pass	name	regdate
next()	MUST	****	성낙현	2021.09.15
next()	HAVE	****	최현우	2021.09.15
:	JSP	****	이복연	2021.09.15
	RABBIT	****	골든레빗	2021.09.15

- int getInt() : 정수형으로 추출
- Date getDate() : 날짜형으로 추출
- String getString() : 문자형으로 추출(가장 많이 사용됨)