

1) Oracle

- [Windows 64-bit with JDK 11 included](#) 다운로드
- 저장 경로, 버전 상관없이 작동
- C:\sqldeveloper-23\sqldeveloper : 바탕화면에 바로가기 만들기

2) Run SQL Command Line 실행

- conn / as sysdba : connected 확인됨
- 사용자 계정에 대한 승인
 - alter user hr account unlock identified by hr; : User altered

```
Copyright (c) 1982, 2014, Oracle. All rights reserved.  
SQL> conn / as sysdba  
Connected.  
SQL> alter user hr account unlock identified by hr;  
User altered.
```

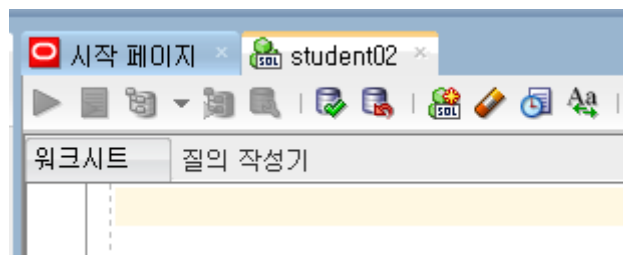
user hr(아이디) / by hr (비번)

- 접속 및 확인
 - conn hr / hr : connected
 - select count(*) from employees;

```
SQL> conn hr  
Enter password:  
Connected.  
SQL> select count(*) from employees;  
  
COUNT(*)  
-----  
107
```

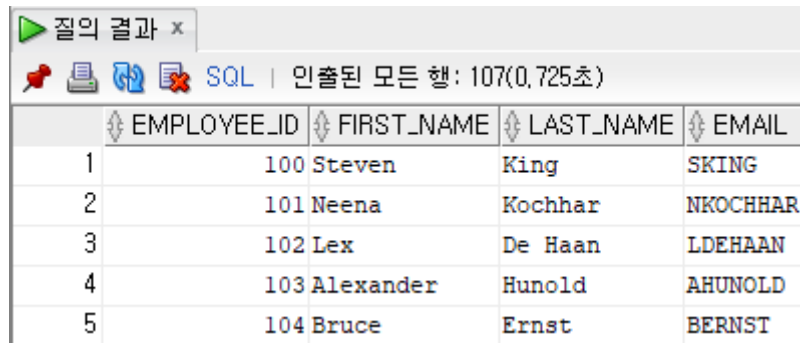
3) sqldeveloper 실행

- 이전 SQL 설치 : 아니오
- 좌측 + : 새 접속
- name 아무거나,
- 이름 비번 : hr hr
- 테스트
- 상태 : 성공



생성된 스키마

- 쿼리문으로 작성
 - `select * from employees;` : ctrl+Enter
 - 이후는 mariadb 와 동일하므로 추가 학습은 배제



질의 결과 x

SQL | 인출된 모든 행: 107(0.725초)

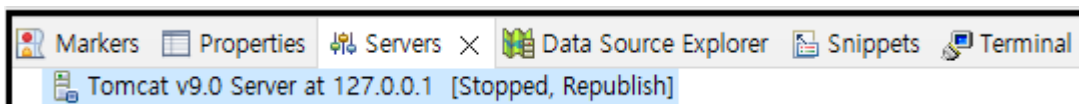
	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL
1	100	Steven	King	SKING
2	101	Neena	Kochhar	NKOCHHAR
3	102	Lex	De Haan	LDEHAAN
4	103	Alexander	Hunold	AHUNOLD
5	104	Bruce	Ernst	BERNST

SQL Command Line 보다 가시적

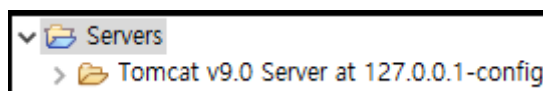
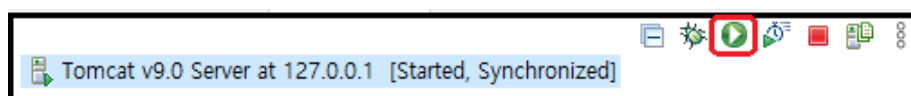
- 작성한 그대로 저장, VS Code 로 호출시 작성한 쿼리문이 확인됨

4) Eclipse IDE

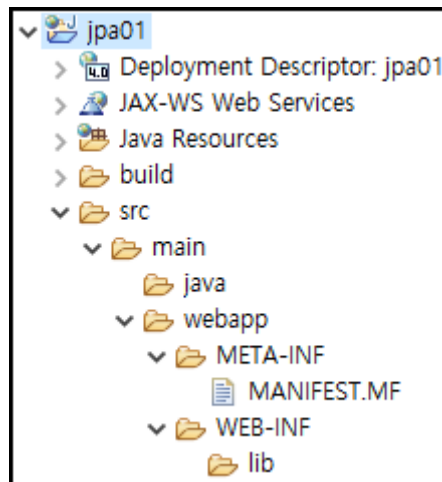
- 다운 경로 : [Eclipse IDE 2023-03 R Packages](#) > [Eclipse IDE for Enterprise Java and Web Developers](#)
- 설치 경로 : C:\Users\student02 에 eclipse 디렉토리 이동
 - 현재 VS Code, IntelliJ, Eclipse 셋다 공존시 에러 발생 높음
 - uninstall 명령어가 없으므로 해당 디렉토리 전체를 삭제할 예정
- eclipse 실행
 - eclipse 가 저장될 경로 : C:\eclipse-workspace
 - VS Code 는 에디터(열기, 편집기능)에 집중
 - eclipse 는 IDE 로 프로젝트 생성이 주목적
- Create a JPA project
 - project Name : 작성과 동시에 디렉토리 생성됨
 - Target runtime : jre
 - JVM : 가상머신, JRE : 개발기능이 빠진 구동, JDK : 모두 포함
 - addconnection : 연결될 DB 설정, Generic JDBC, 현재는 없음, 추후 시행
- Create Dynamic Web Project : JPA 로 웹 프로젝트 생성시 동적 구동
- Tomcat 과 연동 (ctrl + F11)
 - connector project : 채팅 프로그램용
 - File > New > other > server > apache > Tomcat v9.0 Server
 - localhost : 127.0.0.1



- Start

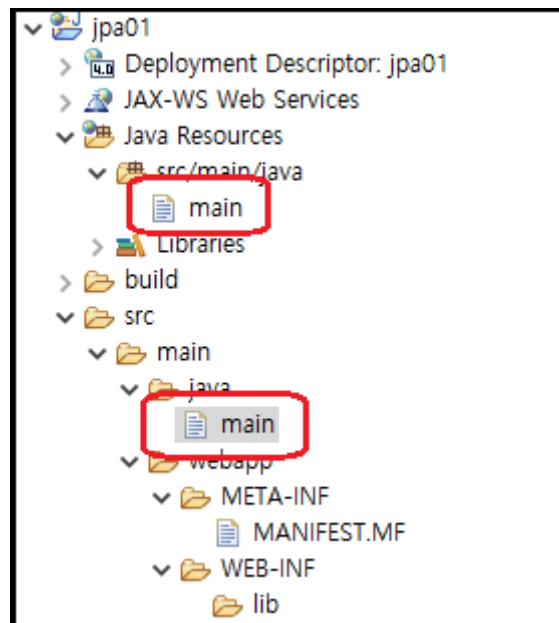


- 디렉토리 별 설명



- src(source)
- webapp > meta-inf : html 의 head 에 해당
- webapp > web-inf : 실제로 웹 앱들이 저장될 공간

- Java 와 Java Resources 자동 연동



- java 의 작업은 둘다 설치 및 연동됨

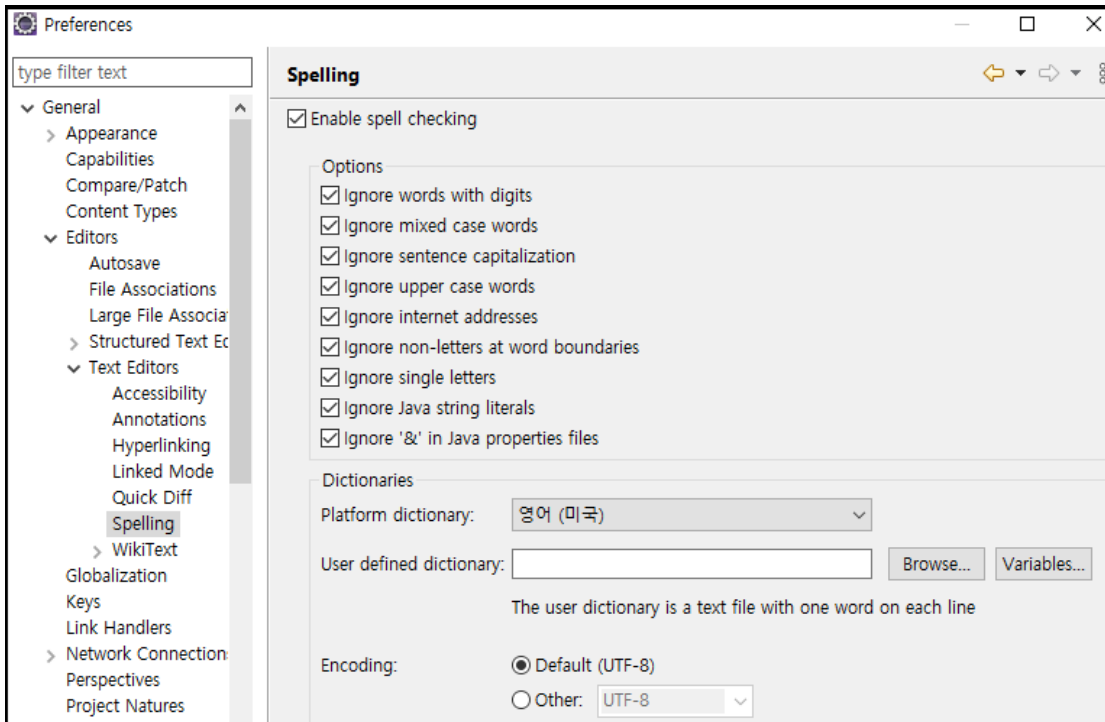
- JSP File 생성시 관련 초기 구문도 자동 생성

```

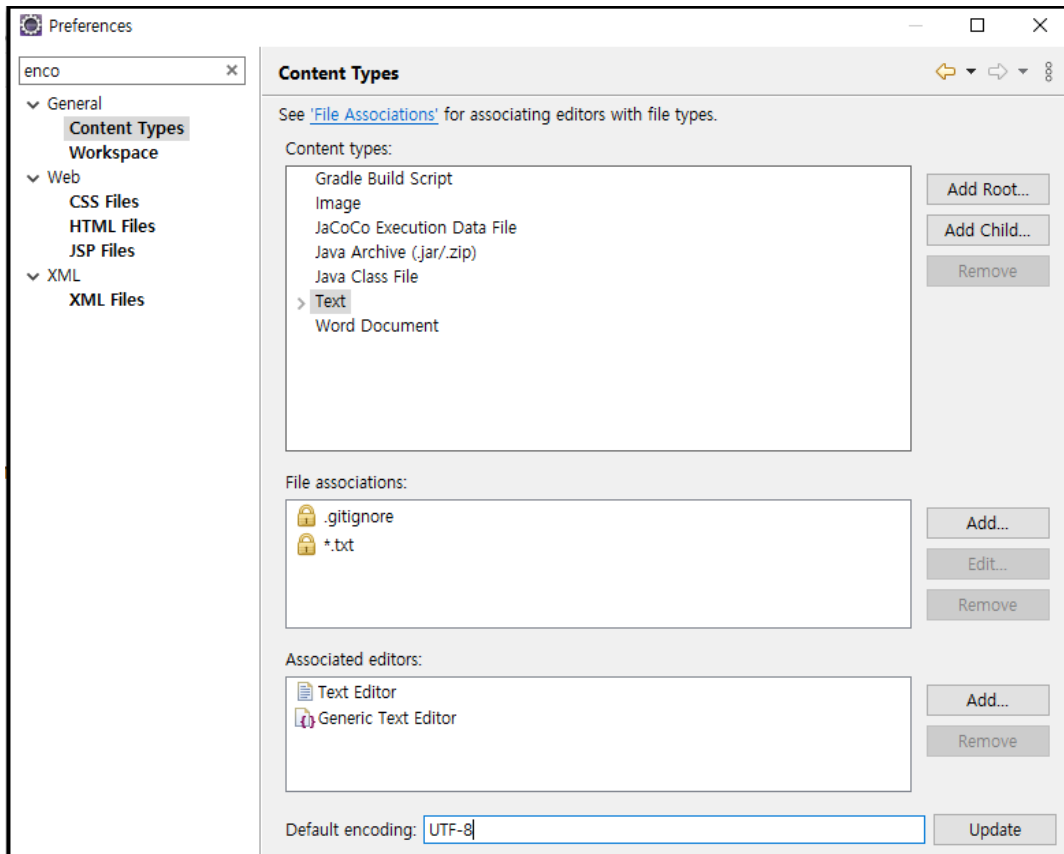
1 <%@ page language="java" contentType="text/html; charset=EUC-KR"
2   pageEncoding="EUC-KR"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="EUC-KR">
7 <title>Insert title here</title>
8 </head>
9 <body>
10
11 </body>
12 </html>

```

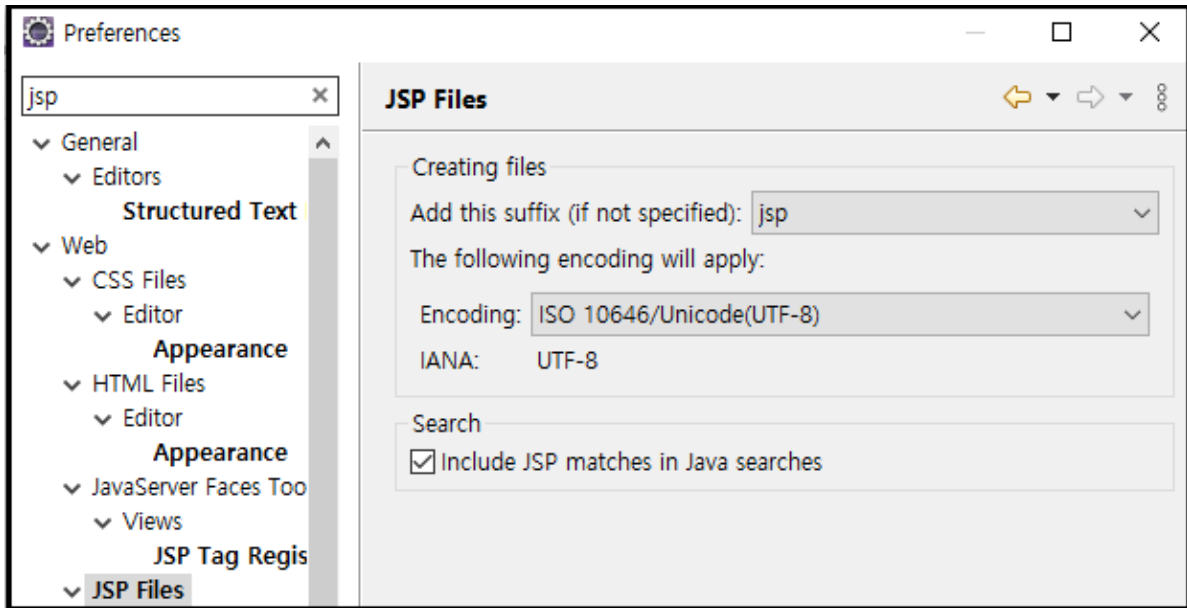
- Maven 과 Graden 사용 예정
- index 가 기준, main 과 default 는 공통 서식 적용시 사용
- UTF-8 설정 : 상단메뉴 > window > preference
 - General > Editors > Text Editors > Spelling : Default(UTF-8)



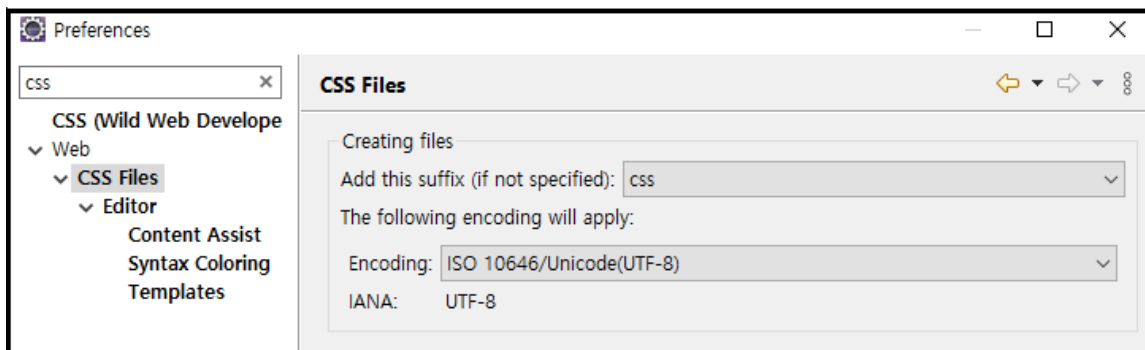
- enco 검색 > Content Types >: Default : (UTF-8)



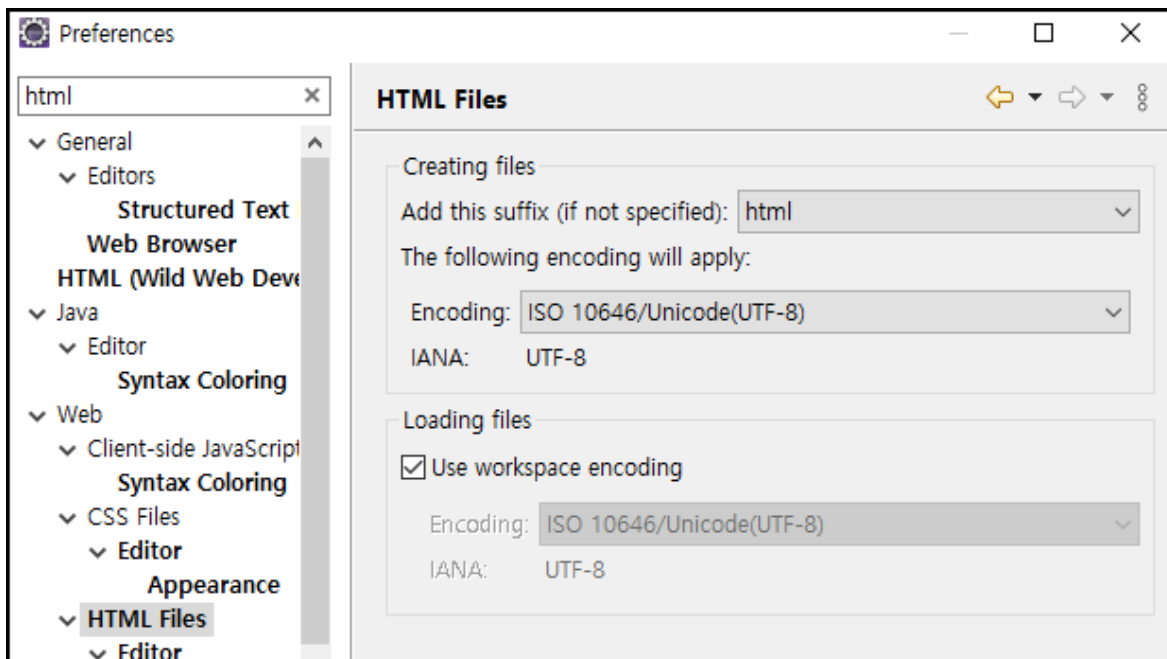
- jsp 검색 > JSP Files : Encoding : UTF-8



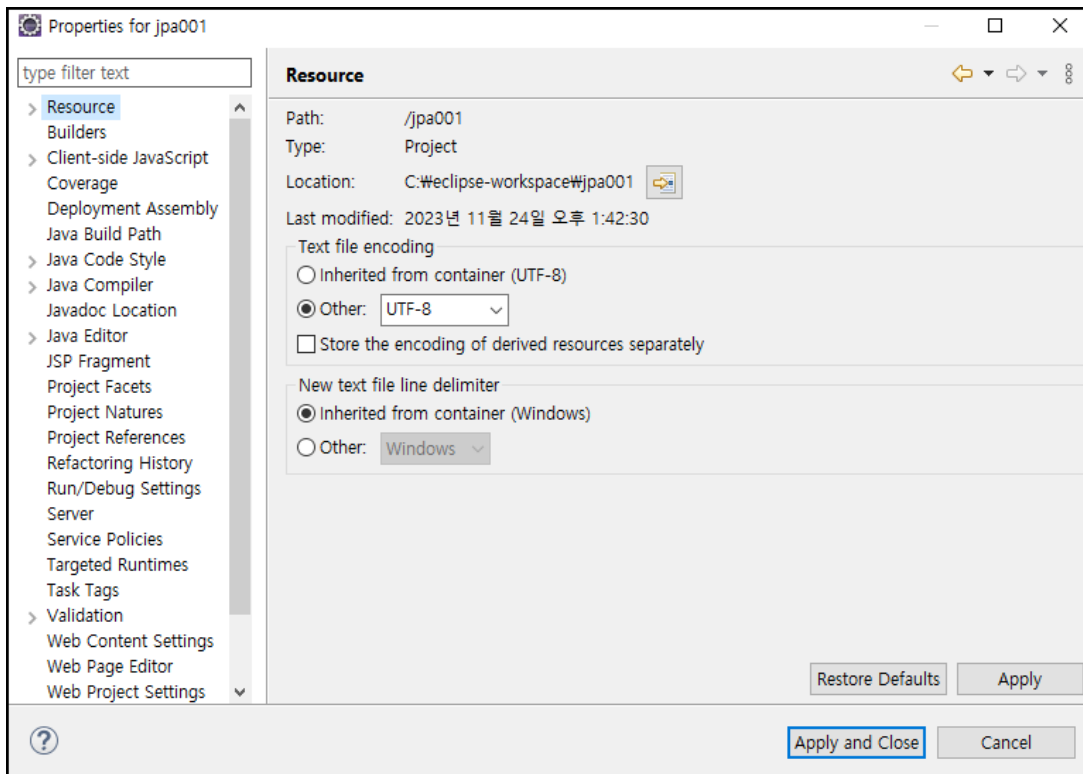
- css 검색 > CSS Files : Encoding : UTF-8



- html 검색 > HTML Files : Encoding : UTF-8



- 전부 update & apply 적용
 - 미리 만들어진 파일(파일에서 우클릭)도 잘 적용되었나 확인 할 것



5) 예제 실습

- WEB-INF > lib > index.jsp 생성 및 작성

쇼핑몰 회원관리 ver1.0

[회원등록](#) [회원목록조회/수정](#) [회원매출조회](#) [홈으로](#)

쇼핑몰회원관리 프로그램

쇼핑몰 회원정보의 회원매출정보 데이터베이스를 구축하고 회원관리 프로그램을 작성하는 프로그램이다.

프로그램 작성 순서

1. 회원정보 테이블을 생성한다.
2. 매출정보 테이블을 생성한다.
3. 회원정보, 매출정보 테이블에 서시된 문제지의 참조데이터를 추가생성한다.
4. 회원정보 입력 화면 프로그램을 작성한다.
5. 회원정보 조회 프로그램을 작성한다.
6. 회원매출정보 조회 프로그램을 작성한다.

HRDKOREA Copyright©2016 All rights reserved. Human Resources Development Service of Korea

5-1) index.jsp

- 오라클 db에서 꺼내라는 함수문
 - 오라클에 만들어진 데이터베이스 파일을 현재 jsp 에서 사용하기 위한 명령어

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ page import = "java.sql.*, javax.sql.*"%>
<%
    Connection con=null;
    PreparedStatement pstmt=null;
    ResultSet rs=null;

    try{
        Class.forName("oracle.jdbc.OracleDriver");
        con=DriverManager.getConnection
            ("jdbc:oracle:thin:@//localhost:1521/xe","system","1234");

        pstmt=con.prepareStatement("select * from member_tbl_02");

        rs=pstmt.executeQuery();
    }
    %>
```

- select * from member_tbl_02 : member_tbl_02 의 모든 것을 가져옴
- jdbc(Java Database Connection) : oracle:thin:@//localhost:1521/xe,"system","1234"
 - 기본 설정을 호출 / system(사용자이름) / 1234 (비밀번호)
 - 초기에는 둘다 한자리에, 이후 외부서버와 DB서버를 따로 설정해야 안전
- rs=pstmt.executeQuery():
 - pstmt 에 executeQuery(): 를 담음
- 자바의 라이브러리 관련 모든 명령어를 추가

```
<%@ page import = "java.sql.*"%>
```

- java 에서 예외처리시

```
</body>
<%
    }catch(Exception e){
    }finally{
        if(rs!=null) rs.close();
        if(pstmt!=null) pstmt.close();
        if(con!=null) con.close();
    }
    %>
</html>
```

- try { 오류/예외발생 가능한 코드 } : 가장 위
 - { html } 이 위치하도록
 - catch (Type1) { 1번 유형의 예외처리 코드 } : body 이후
 - 현재는 {} 안이 무조건이라, 예외발생해도 특정 행동 없음
 - finally { 예외가 발생or미발생시 실행되는 코드 } : body 이후, catch 와 함께
- WEB-INF > index.jsp 파일 이동
 - 상위 디렉토리인 jpa001 > Run As > Run on Server > 클릭 > next > jpa001 > 구동

5-2) memberForm.jsp

- JSP 문은 <%%> 의 경계가 다소 불분명

```

</tr>
<%
    if(rs.next()){
        do{
            String grade="직원";
            if(rs.getString(6).equals("A")) grade="VIP";
            else if(rs.getString(6).equals("B")) grade="일반";
        }
    }
    %>
<tr>
<td><a href="modForm.jsp?custno=<%=rs.getString(1)%>"><%=rs.getString(1) %></a></td>
<td><%=rs.getString(2) %></td>
<td><%=rs.getString(3) %></td>
<td><%=rs.getString(4) %></td>
<td><%=rs.getString(5).substring(0,10) %></td>
<td><%=grade %></td>
<td><%=rs.getString(7) %></td>
</tr>
<%
    }while(rs.next());
    }else{
    }
    %>
</section>

```

- <% if do else if %> 에서 한번 종료, <% while, else 에서도 다시 %>
 - (next blank space) : 빈칸
- modForm 에서도 사용됨

5-3) joinForm

- auto increment

```

int custno=100001;
if(rs.next()){
    custno=rs.getInt(1)+1;
}

```

- checkForm(this) : 체크버튼 눌렀을시 원래대로

```

<form name="f" action="joinProcess.jsp" method="post" onsubmit="return checkForm(this)">

```

- post : update

5-4) modForm

- pstmt.setString(1,custno);
 - custno 는 원래 >=100 이런식인데, 현재는 ? 은 무엇이 올지 모름
 - 하나의 열로 custno을 받아라

5-5) saleForm

- SQL의 최대 구문 : select 필드 from 테이블 where 조건 group by 묶음 order by 정렬

```
String sql="select m1.custno, custname, grade, sum(price)"
        +" from MEMBER_TBL_02 m1 inner join money_tbl_02 m2"
        +" on m1.custno=m2.custno"
        +" group by m1.custno, m1.custname, m1.grade order by sum(price) desc";
```

회원매출조회

회원번호	회원성명	고객등급	매출
100001	김행복	VIP	8000
100004	최사랑	VIP	4900
100003	장밀음	일반	3000
100002	이죽복	일반	2500

- + 로 한줄이라는 의미
- 여러 필드에 떨어진 조항들을 한번에 호출가능
- select m1의 custno 를(PK가능),
 - custname 은 from 에서 join 될 양 테이블에 있거나, m1.custno의 종속된 필드
- inner join : 하나로 묶어서 호출하되 뒤에것은 뒤로 / outer join : 기준 필드의 왼쪽
- group by : 작성된 순서대로
- order by : desc 내림차순으로 (큰값을 위로) <-> asc