

23-11-27

- 오라클 사용하여 SQL 데이터 생성
- SQL INNER JOIN

2) 데이터 입출력 요건에 맞게 투표 이력 테이블, 후보자 테이블, 정당코드 테이블을 생성하시오.
가) 투표이력테이블 명세서 (테이블 명 : TBL_VOTE_202005)

순서	컬럼ID	컬럼명	형태	길이	NULL	비고
1	V_IUMIN	주민번호	CHAR	13	NOT NULL	PRIMARY KEY
2	V_NAME	성명	VARCHAR2	20		
3	M_NO	후보번호	CHAR	1		
4	V_TIME	투표시간	CHAR	4		
5	V_ARE	투표장소	CHAR	20		
6	V_CONFIRM	유권자확인	CHAR	1		

[투표이력테이블 샘플 데이터]

주민번호	성명	후보번호	투표시간	투표장소	유권자확인
99010110001	김유권	1	0930	제1투표장	N
89010120002	이유권	2	0930	제1투표장	N
69010110003	박유권	3	0930	제1투표장	Y
59010120004	홍유권	4	0930	제1투표장	Y
79010110005	조유권	5	0930	제1투표장	Y
89010120006	최유권	1	0930	제1투표장	Y
59010110007	지유권	1	0930	제1투표장	Y
49010120008	장유권	3	0930	제1투표장	Y
79010110009	전유권	3	0930	제1투표장	Y
89010120010	강유권	4	0930	제1투표장	Y
99010110011	신유권	5	0930	제1투표장	Y
79010120012	오유권	1	1330	제1투표장	Y
69010110013	현유권	4	1330	제2투표장	Y
89010110014	황유권	2	1330	제2투표장	Y
99010110015	유유권	3	1330	제2투표장	Y
79010110016	한유권	2	1330	제2투표장	Y
89010110017	윤유권	4	1330	제2투표장	Y
99010110018	양유권	2	1330	제2투표장	Y
99010110019	구유권	4	1330	제2투표장	Y
79010110020	황유권	5	1330	제2투표장	Y
69010110021	백유권	3	1330	제2투표장	Y
79010110022	전유권	3	1330	제2투표장	Y
99010110023	고유권	1	1330	제2투표장	Y
59010110024	권유권	3	1330	제2투표장	Y

나) 후보자테이블 명세서 (테이블 명 : TBL_MEMBER_202005)

순서	컬럼ID	컬럼명	형태	길이	NULL	비고
1	M_NO	후보번호	CHAR	1	NOT NULL	PRIMARY KEY
2	M_NAME	성명	VARCHAR2	20		
3	P_CODE	소속정당	CHAR	2		
4	P_SCHOOL	최종학력	CHAR	1		
5	M_IUMIN	주민번호	CHAR	13		
6	M_CITY	지역구	VARCHAR2	20		

[후보자테이블 샘플 데이터]

후보번호	성명	소속정당	최종학력	주민번호	지역구
1	김후보	P1	1	6603011999991	수선화동
2	이후보	P2	3	5503011999992	민들레동
3	박후보	P3	2	7703011999993	나팔꽃동
4	조후보	P4	2	8803011999994	진달래동
5	최후보	P5	3	9903011999995	개나리동

다) 정당코드테이블 명세서 (테이블 명 : TBL_PARTY_202005)

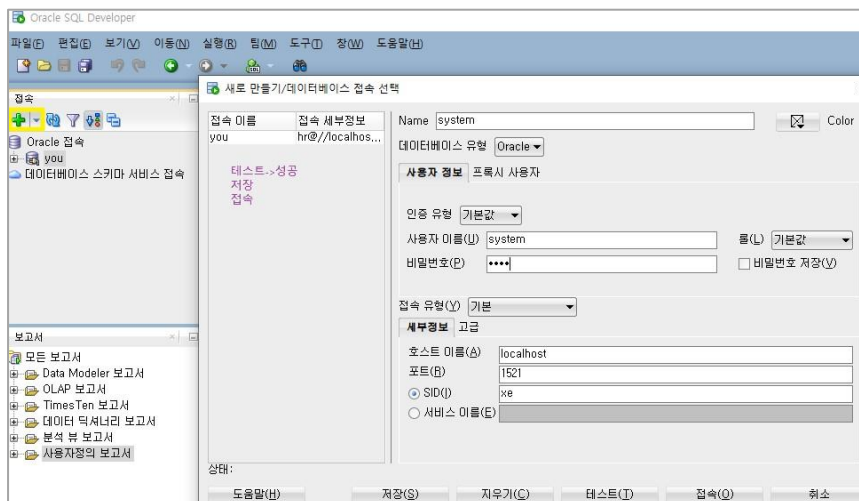
순서	컬럼ID	컬럼명	형태	길이	NULL	비고
1	P_CODE	정당코드	CHAR	2	NOT NULL	Primary key
2	P_NAME	명칭	VARCHAR2	20		
3	P_INDATE	등록연월일	DATE			
4	P_READER	대표자	VARCHAR2	20		
5	P_TEL1	전화번호1	CHAR	3		
6	P_TEL2	전화번호2	CHAR	4		
7	P_TEL3	전화번호3	CHAR	4		

[정당코드테이블 샘플 데이터]

정당코드	명칭	등록연월일	대표자	전화번호1	전화번호2	전화번호3
P1	A정당	2010-01-01	위대표	02	1111	0001
P2	B정당	2010-02-01	별대표	02	1111	0002
P3	C정당	2010-03-01	기대표	02	1111	0003
P4	D정당	2010-04-01	옥대표	02	1111	0004
P5	E정당	2010-05-01	임대표	02	1111	0005

3) 후보자 테이블, 정당코드 테이블의 데이터는 SQL 문장을 사용하여 생성된 테이블에 데이터를 입력하시오.
4) 투표이력 테이블의 데이터는 투표이력 샘플 데이터를 참조하여 투표하기 화면에서 입력하시오.
5) 화면별 업무 요구사항 및 화면 구성 요건에 맞게 화면을 구현하시오.
[참고사항]
- 화면의 구성요소는 필수사항이다.
- 화면의 색깔, 폰트 등 스타일 구성요소는 선택 사항이다.

Oracle SQL Developer (바탕화면에 바로가기 만들었던 것)를 사용하여 sql을 만든다.



하나 새로 만들고 테스트, 저장, 접속 하면 sql문 만들 수 있는 화면에 접속이 가능하다

테이블 만들기 -> create

Oracle SQL Developer : system

```

create table tbl_vote_202005 (
  V_jumin char(13) not null primary key,
  V_name varchar(20),
  V_no char(1),
  V_time char(4),
  V_area char(20),
  V_confirm char(1)
);

```

작업이 완료되었습니다. (0.094초)

오류 보고 -

ORA-00904: invalid identifier
00904. 00000 - "ks: invalid identifier"
*Cause:
*Action:

Table TBL_VOTE_2020050I (가) 생성되었습니다.

박원일 2023-11-27

2) 데이터 입출력 요건에 맞게 투표 이력 테이블, 후보자 테이블, 정당코드 테이블을 생성하시오.
가) 투표이력테이블: 박원일, 역 - TBL_VOTE_202005

순서	회원ID	주민번호	성명	성별	생년월일	투표번호	투표시간	투표장소	투표방법
1	V_JUMIN	CHAR	13	NOT NULL	PRIMARY KEY				
2	V_NAME	VARCHAR2	20						
3	V_NO	CHAR	1						
4	V_TIME	CHAR	4						
5	V_AREA	CHAR	20						
6	V_CONFIRM	CHAR	1						

[후보자데이터를 참조할 데이터]

주민번호	성명	투표번호	투표시간	투표장소	투표방법
99010110001	김우현	1	0930	제1투표장	N
99010110002	이우현	2	0930	제1투표장	N
99010110003	박우현	3	0930	제1투표장	Y
99010110004	홍우현	4	0930	제1투표장	Y
99010110005	조우현	5	0930	제1투표장	Y
99010110006	최우현	1	0930	제1투표장	Y
99010110007	지우현	1	0930	제1투표장	Y
99010110008	최우현	1	0930	제1투표장	Y

tbl_vote_202005 TABLE

Oracle SQL Developer : system

```

CREATE TABLE tbl_member_202005 (
  M_no char(1) not null primary key,
  M_name varchar(20),
  P_code char(2),
  P_school char(1),
  M_jumin char(13),
  M_city varchar(20)
);

```

작업이 완료되었습니다. (0.035초)

오류 보고 -

ORA-00904: invalid identifier
00904. 00000 - "ks: invalid identifier"
*Cause:
*Action:

Table TBL_MEMBER_2020050I (가) 생성되었습니다.

Table TBL_MEMBER_2020050I (가) 생성되었습니다.

나) 후보자데이터를 참조할 테이블 명 : TBL_MEMBER_202005

순서	회원ID	주민번호	성명	성별	생년월일	투표번호	투표시간	투표장소	투표방법
1	M_NO	CHAR	1	NOT NULL	PRIMARY KEY				
2	M_NAME	VARCHAR2	20						
3	P_CODE	CHAR	2						
4	P_SCHOOL	CHAR	1						
5	M_JUMIN	CHAR	13						
6	M_CITY	CHAR	20						

[후보자데이터를 참조할 데이터]

주민번호	성명	투표번호	투표시간	투표장소	투표방법
1	김우현	P1	1	6603011999991	수선화동
2	이우현	P2	2	5503011999992	민들레동
3	박우현	P3	2	7703011999993	나물골동
4	홍우현	P4	2	8803011999994	전달대동
5	최우현	P5	3	9903011999995	개나리동

tbl_member_202005 TABLE

Oracle SQL Developer : system

```

CREATE TABLE tbl_party_202005 (
  P_code char(2) not null PRIMARY KEY,
  P_name varchar(20),
  P_indate date,
  P_reader varchar(20),
  P_tel1 char(3),
  P_tel2 char(4),
  P_tel3 char(4)
);

```

작업이 완료되었습니다. (0.038초)

오류 보고 -

ORA-00955: name is already used by an existing object
00955. 00000 - "name is already used by an existing object"
*Cause:
*Action:

Table TBL_PARTY_2020050I (가) 생성되었습니다.

다) 정당코드데이터를 참조할 테이블 명 : TBL_PARTY_202005

순서	회원ID	주민번호	성명	성별	생년월일	투표번호	투표시간	투표장소	투표방법
1	P_CODE	CHAR	2	NOT NULL	Primary key				
2	P_NAME	VARCHAR2	20						
3	P_INDATE	DATE							
4	P_READER	CHAR	20						
5	P_TEL1	CHAR	3						
6	P_TEL2	CHAR	4						
7	P_TEL3	CHAR	4						

[후보자데이터를 참조할 데이터]

정당코드	정당명	등록연월일	대표자	전화번호1	전화번호2	전화번호3
P1	A정당	2010-01-01	김대표	02	1111	0001
P2	B정당	2010-02-01	박대표	02	1111	0002
P3	C정당	2010-03-01	최대표	02	1111	0003
P4	D정당	2010-04-01	정대표	02	1111	0004
P5	E정당	2010-05-01	임대표	02	1111	0005

3) 후보자 테이블, 정당코드 테이블의 데이터는 SQL 문장을 사용하여 생성된 테이블을 데이터로 입력하시오.
4) 투표이력 테이블의 데이터는 투표이력 생성 데이터를 참조하여 투표하기 화면에서 입력하시오.
5) 화면의 입부 요구사항 및 화면 구성 요건에 맞게 화면을 구현하시오.

[참고사항]

- 화면의 구성요소는 필수사항이다.
- 화면의 색상, 폰트 등 스타일 구성요소는 선택 사항이다.

tbl_party_202005 TABLE

테이블 만드는 것, 튜플이름, 형태, 길이 null값 유무, 프라이머리키 지정 등 위에 자료를 참고하여 테이블을 만들어줬다 (테이블 3개)

- 내 시스템 안에 있는 모든테이블 출력하는 구문

```
select * from all_tables;
```

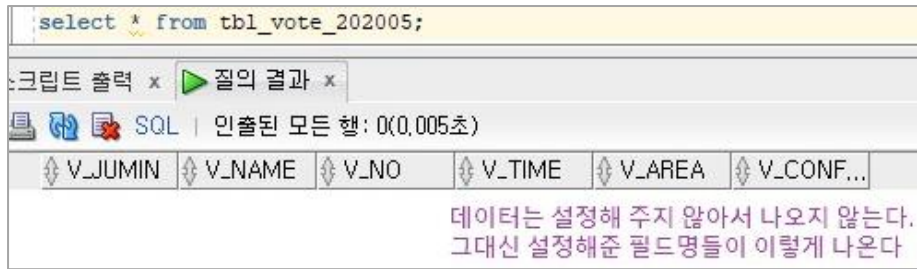
내 시스템 안에 있는 모든 테이블

스크립트 출력 x 실행 결과 x

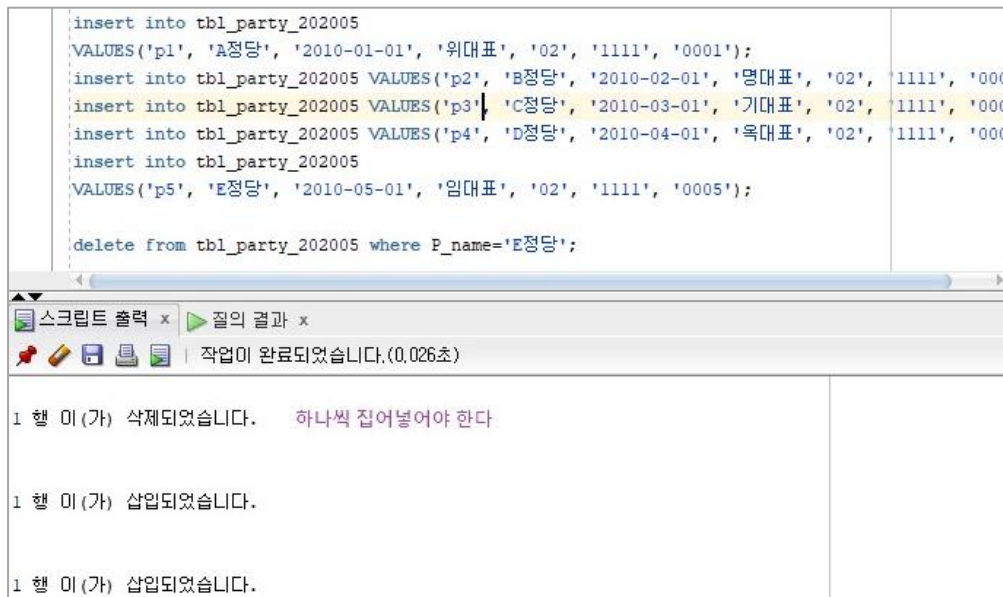
SQL | 50개의 행이 인쇄됨(0.712초)

	OWNER	TABLE_NAME	TABLESPACE_NAME	CLUSTER_NAME	IOT_NAME	STATUS	PCT_FREE
1	SYS	ICOL\$	SYSTEM	C_OBJ\$	(null)	VALID	0
2	SYS	CON\$	SYSTEM	(null)	(null)	VALID	10
3	SYS	UNDO\$	SYSTEM	(null)	(null)	VALID	10
4	SYS	PROXY_ROLE_DATA\$	SYSTEM	(null)	(null)	VALID	10
5	SYS	FILE\$	SYSTEM	(null)	(null)	VALID	10
6	SYS	UET\$	SYSTEM	C_FILE\$_BLOCK\$	(null)	VALID	0
7	SYS	IND\$	SYSTEM	C_OBJ\$	(null)	VALID	0
8	SYS	SEG\$	SYSTEM	C_FILE\$_BLOCK\$	(null)	VALID	0

실제 데이터 집어넣어야 한다 -> insert



테이블 형태와 조건들을 정해서 만들어졌지만, 출력할 때 데이터들이 들어있지 않아서 출력되지 않는다 -> INSERT로 데이터를 넣어줄거다.



행은 하나씩 CTRL+ENTER로 삽입해줘야한다.

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

INSERT사용방법처럼 테이블 이름과 열이름들에 각 맞는 값을 넣어준다(열 이름은 생략가능)



```
--삭제
delete from tbl_party_202005 where P_name='E정당';
drop TABLE tbl_member_202005;
```

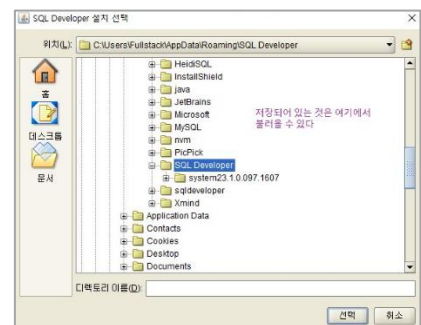
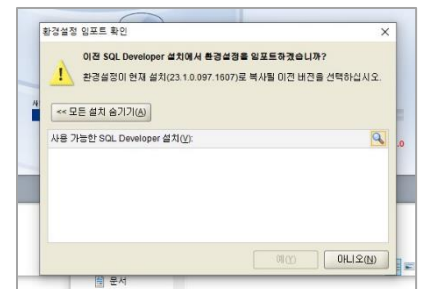
삭제하는 방법 delete로 하거나, drop으로 하면된다.



SQL developer에 저장되어 있는 주소이다. 여기에 있는 것을 다 지워도 다시 실행하면 데이터가 남아있다.

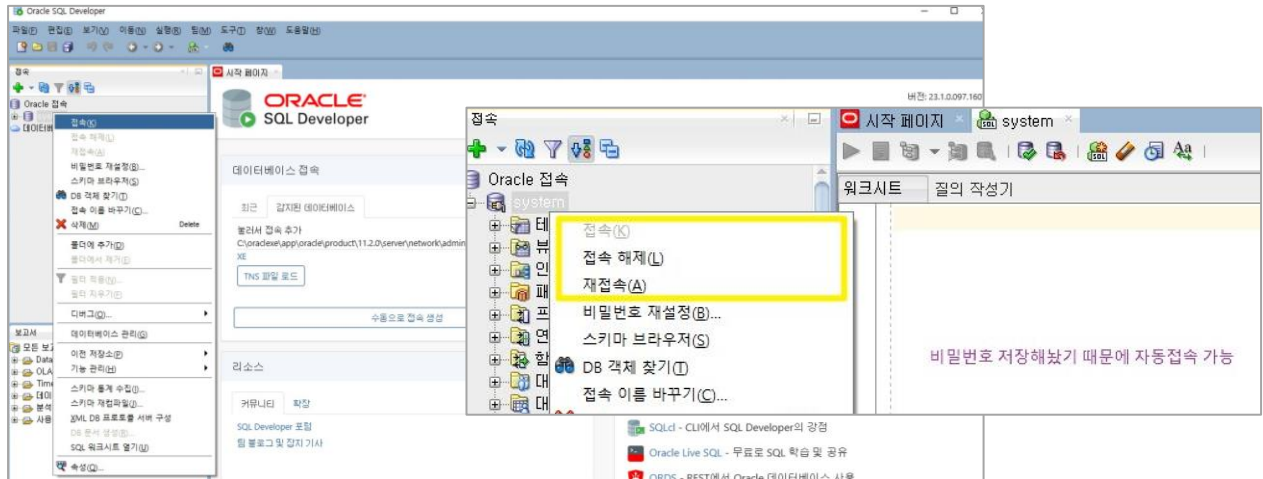


오라클DB는 시스템 자체가 크다



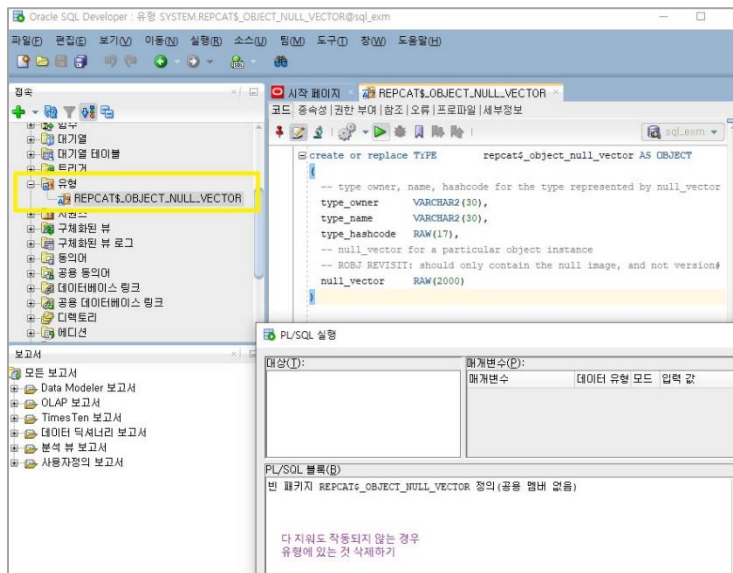
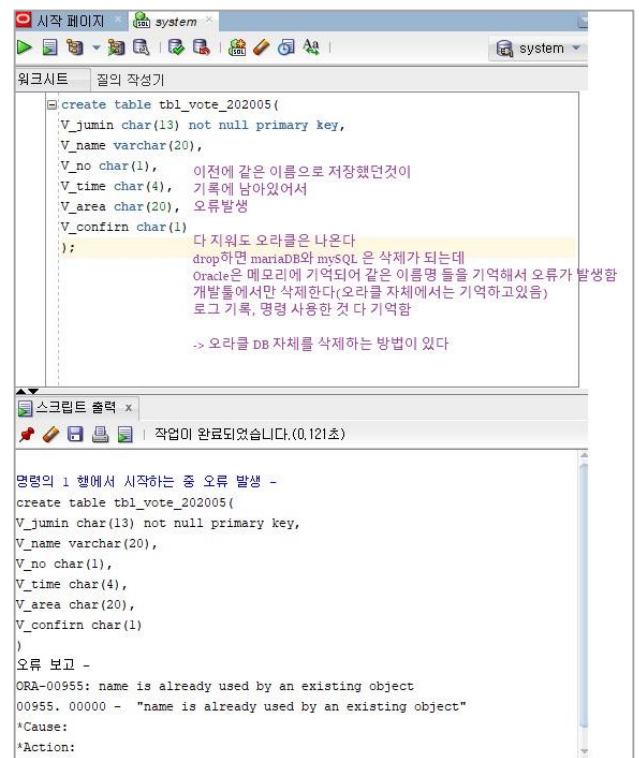
오라클에서 만들 때 실수를 용납하지 않는다 메모장에 넣고 검토한 다음 시스템 돌리는 것이 좋다

오라클 정보 보관 => 은행과 정부기관에서 사용 삭제한 기록도 남아있다



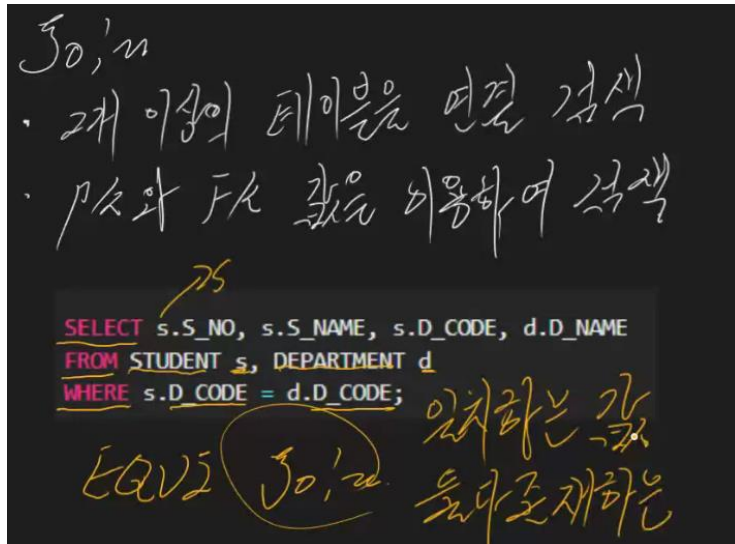
Oracle SQL developer 접속 하면 구문 입력할 수 있다.

저장하면 .sql 스크립트로 저장됨



개념

Join : 2개 이상의 테이블을 연결 검색할 때 사용
PK와 FK의 값을 이용하여 검색한다



Join_exm(oracle)

S_NO	S_NAME	D_CODE	S_SENIOR	S_GRADE
1	15 김가람	com	(null)	83
2	16 정다솜	han	(null)	96
3	17 이진수	com	15	95
4	19 황명훈	han	16	75
5	37 김아름	(null)	17	55

student테이블

D_CODE	D_NAME
1 com	컴퓨터
2 han	국어
3 eng	영어

department테이블

#방법1

```
SELECT s.S_NO, s.S_NAME, s.D_CODE, d.D_NAME FROM STUDENT s, DEPARTMENT d WHERE s.D_CODE = d.D_CODE;
```

두개의 테이블을 하나의 테이블로 만드는 명령어

FROM student를 s로 간단하게 쓴 것, department는 d로 간단하게 쓴 것.

그래서 s의 s_NO와 그 외의 것을 찾겠다, 조건은 s와 d의 D_CODE가 같을 때!!!

Where s.D_CODE = d.D_CODE;

Equi join : 일치하는 값, 둘다 존재하는 값을 찾는 것이다

(null값있으면 값을 가져올 수 없으므로 김아름은 가져올수없다)

S_NO	S_NAME	D_CODE	D_NAME
1	15 김가람	com	컴퓨터
2	16 정다솜	han	국어
3	17 이진수	com	컴퓨터
4	19 황명훈	han	국어

#방법2

```
SELECT s.S_NO, s.S_NAME, s.D_CODE, d.D_NAME FROM STUDENT s inner join DEPARTMENT d on s.D_CODE = d.D_CODE;
```

코드값 같을 때만 출력한다 코드의 null값이 들어있는 김아름은 출력하지 않는다(D_CODE기준)

다른 튜플값이 NULL인 경우는 D_CODE에 포함되어 있으면 출력이 된다

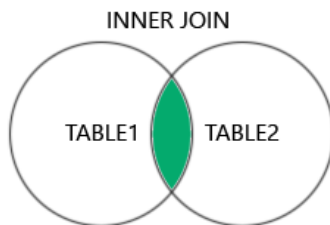
둘다 존재할 때 출력한다, 위에 where 구문과 같은 구문이다

inner join을 더 많이 사용한다.

```
SELECT s.S_NO, s.S_NAME, s.S_SENIOR, s.D_CODE, d.D_NAME FROM STUDENT s inner join DEPARTMENT d on s.D_CODE = d.D_CODE;
```

위 구문에서 s.S_SENIOR 만 추가해 줬는데 NULL값이 있는데도 출력했다

왜냐, 기준이 D_CODE이기 때문이다.



INNER JOIN(일치하는 값)

	S_NO	S_NAME	S_SENIOR	D_CODE	D_NAME
1	15	김가람	(null)	com	컴퓨터
2	16	정다솜	(null)	han	국어
3	17	이진수	15	com	컴퓨터
4	19	황명훈	16	han	국어

```
SELECT s.S_NO, s.S_NAME, s.S_SENIOR, s.D_CODE, d.D_NAME
FROM STUDENT s inner join DEPARTMENT d on s.D_CODE = d.D_CODE
order by s.s_no asc, s.S_NAME desc;
```

스크립트 출력 x | 질의 결과 x

SQL | 인출된 모든 행: 4(0.004초)

	S_NO	S_NAME	S_SENIOR	D_CODE	D_NAME
1	15	김가람	(null)	com	컴퓨터
2	16	정다솜	(null)	han	국어
3	17	이진수	15	com	컴퓨터
4	19	황명훈	16	han	국어

위 구문에 order by를 추가하여 출력해준 값이다.

asc : 오름차순 정렬 (no값)

desc : 내림차순값 => no값이 같을 경우에 name은 내림차순으로 정렬하라는 것

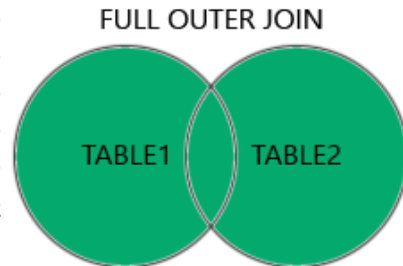
(여러 개가 조건이 되는 경우 맨 왼쪽이 기준이 된다.(여기서는 NO가 먼저 정렬됨 그리고 나서 같은 값이 존재할 경우 name이 정렬되는 것)

```
SELECT s.S_NO, s.S_NAME, s.S_SENIOR, s.D_CODE, d.D_NAME
FROM STUDENT s full outer join DEPARTMENT d on s.D_CODE = d.D_CODE;
```

크립트 출력 x 질의 결과 x

SQL | 인출된 모든 행: 6(0.005초)

S_NO	S_NAME	S_SENIOR	D_CODE	D_NAME
1	15 김가람	(null)	com	컴퓨터
2	16 정다솜	(null)	han	국어
3	17 이진수	15	com	컴퓨터
4	19 황명훈	16	han	국어
5	37 김아름	17	(null)	(null)
6	(null) (null)	(null)	(null)	영어



FULL OUTER JOIN(다 출력한다)

두개의 테이블 모든 값을 포함하여 출력한 것.

-김아름에 D_CODE와 NAME에는 NULL 값으로 NULL이 출력되었고

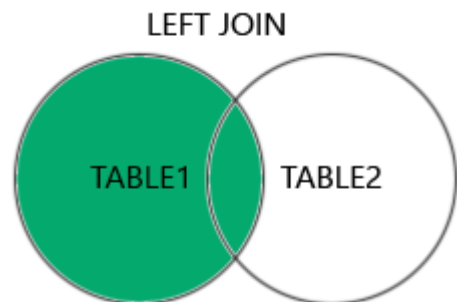
-D_NAME중에 영어는 다른테이블에 해당하는 값이 없으니까 NULL로 출력되었다.

```
SELECT s.S_NO, s.S_NAME, s.S_SENIOR, s.D_CODE, d.D_NAME
FROM STUDENT s left outer join DEPARTMENT d on s.D_CODE = d.D_CODE;
```

크립트 출력 x 질의 결과 x

SQL | 인출된 모든 행: 5(0.006초)

S_NO	S_NAME	S_SENIOR	D_CODE	D_NAME
1	17 이진수	15	com	컴퓨터
2	15 김가람	(null)	com	컴퓨터
3	19 황명훈	16	han	국어
4	16 정다솜	(null)	han	국어
5	37 김아름	17	(null)	(null)



LEFT OUTER JOIN : 기준이 왼쪽이다 (즉, STUDENT가 기준)

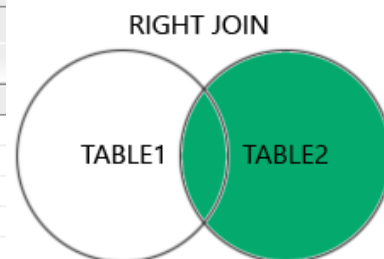
STUDENT와 DEPARTMENT 같은 부분과 STUDENT에만 포함된 부분 가지고 오는 것
오른쪽의 NULL값이 있어도 왼쪽, 즉 STUDENT에 있는 값은 다 가지고 오는 것.

```
SELECT s.S_NO, s.S_NAME, s.S_SENIOR, s.D_CODE, d.D_NAME
FROM STUDENT s right outer join DEPARTMENT d on s.D_CODE = d.D_CODE;
```

크립트 출력 x 질의 결과 x

SQL | 인출된 모든 행: 5(0.005초)

S_NO	S_NAME	S_SENIOR	D_CODE	D_NAME
1	15 김가람	(null)	com	컴퓨터
2	16 정다솜	(null)	han	국어
3	17 이진수	15	com	컴퓨터
4	19 황명훈	16	han	국어
5	(null) (null)	(null)	(null)	영어



RIGHT OUTER JOIN은 기준이 오른쪽이다 (즉, DEPARTMENT가 기준)

왼쪽 테이블이 NULL값이 있어도 오른쪽 테이블에 포함된 값이면 출력하는 것

<pre> SELECT s.S_NO, s.S_NAME, s.S_SENIOR, s.D_CODE, d.D_NAME FROM STUDENT s, DEPARTMENT d where s.D_CODE(+) = d.D_CODE; </pre>				
<div> 스크립트 출력 x 실행 결과 x </div> <div> SQL 인출된 모든 행: 5(0.002초) </div>				
S_NO	S_NAME	S_SENIOR	D_CODE	D_NAME
1	15 김가람	(null)	com	컴퓨터
2	16 정다솜	(null)	han	국어
3	17 이진수	15	com	컴퓨터
4	19 황명훈	16	han	국어
5	(null) (null)	(null)	(null)	영어

RIGHT OUTER JOIN과 같은 값이 출력됨

where조건을 추가하고 +=을 해준 것은 오른쪽 테이블의 값을 왼쪽테이블에 추가해 주겠다는 것

예제

2) 데이터 입출력 요건에 맞게 강사정보, 회원정보, 수강정보 테이블을 생성하시오.					
가) 강사정보 (테이블 명 : TBL_TEACHER_202201)					
순서	컬럼ID	컬럼명	형태	길이	NULL
1	TEACHER_CODE	강사코드	CHAR	3	NOT NULL
2	TEACHER_NAME	강사명	VARCHAR2	15	
3	CLASS_NAME	강의명	VARCHAR2	20	
4	CLASS_PRICE	수강료	NUMBER	8	
5	TEACHER_REGIST_DATE	강사자격취득일	VARCHAR2	8	
[강사정보 샘플 데이터]					
강사코드	강사명	강의명	수강료	강사자격취득일	
100	이초급	초급반	100000	20220101	
200	김중급	중급반	200000	20220102	
300	박고급	고급반	300000	20220103	
400	정상화	심화반	400000	20220104	
나) 회원정보 (테이블 명 : TBL_MEMBER_202201)					
순서	컬럼ID	컬럼명	형태	길이	NULL
1	C_NO	회원번호	CHAR	5	NOT NULL
2	C_NAME	회원명	VARCHAR2	15	
3	PHONE	전화번호	VARCHAR2	11	
4	ADDRESS	주소	VARCHAR2	50	
5	GRADE	등급	VARCHAR2	6	
[회원정보 샘플 데이터]					
회원번호	회원명	전화번호	주소	등급	
10001	홍길동	01011112222	서울시 강남구	일반	
10002	장발장	01022223333	성남시 분당구	일반	
10003	임꺽정	01033334444	대전시 유성구	일반	
20001	성준황	01044445555	부산시 서구	VIP	
20002	이몽룡	01055556666	대구시 북구	VIP	
다) 수강정보 (테이블 명 : TBL_CLASS_202201)					
순서	컬럼ID	컬럼명	형태	길이	NULL
1	REGIST_MONTH	수강월	VARCHAR2	6	NOT NULL
2	C_NO	회원번호	CHAR	5	NOT NULL
3	CLASS_AREA	강의장소	VARCHAR2	15	
4	TUITION	수강료	NUMBER	8	
5	TEACHER_CODE	강사코드	CHAR	3	
[수강정보 샘플 데이터]					
수강월	회원번호	강의장소	수강료	강사코드	
202203	10001	서울본원	100000	100	
202203	10002	성남본원	100000	100	
202203	10003	대전본원	200000	200	
202203	20001	부산본원	150000	300	
202203	20002	대구본원	200000	400	

3개의 테이블을 만들고, 데이터도 집어넣기

테이블 수정이 필요할 때 alter로 변경해주기
아래와 같이 변경해주었다.
primary key 일때 같은 값을 집어 넣게되면
작동하지 않는다

2. 테이블 수정 (ALTER)

▷ 구문

컬럼 이름 수정 : ALTER TABLE [테이블 이름] RENAME COLUMN [변경전 이름] TO [변경 후 이름]
컬럼 추가 : ALTER TABLE [테이블 이름] ADD ([컬럼명][타입][[사이즈]]);
컬럼 수정 : ALTER TABLE [테이블 이름] MODIFY ([수정할 컬럼명][수정할 타입][[DEFAULT 관련]][NOT NULL]);
> DEFAULT 나 NOT NULL 은 생략가능...
컬럼 삭제 : ALTER TABLE [테이블 이름] DROP COLUMN ([컬럼명]);

하나의 테이블의 2개의 primary key가 존재할 수 없다.

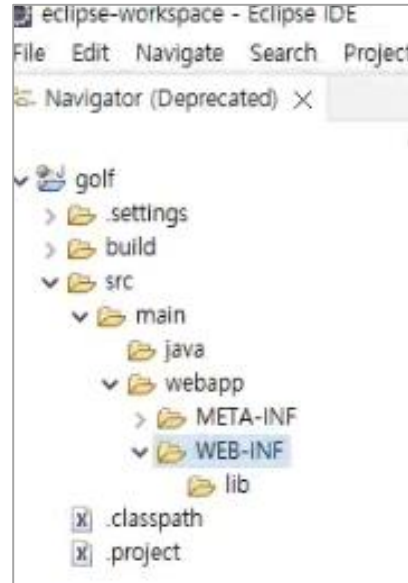
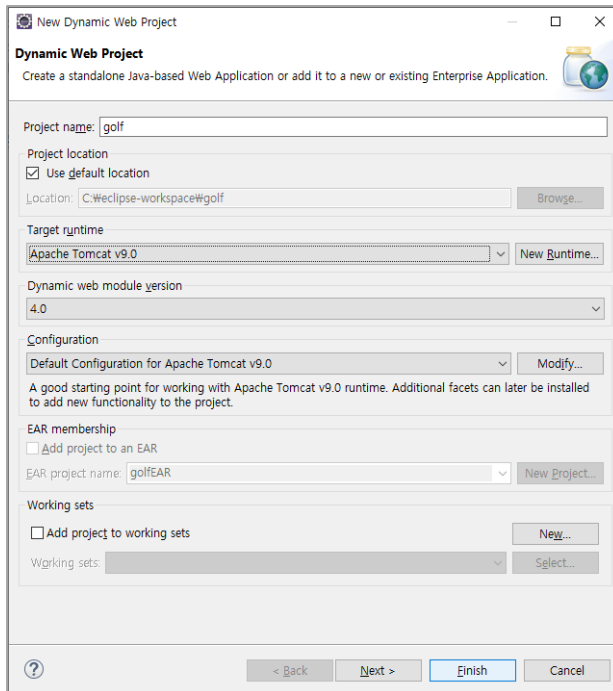
```
alter table tbl_class_202201 modify (regist_month varchar2(6));
```

```
alter table tbl_class_202201 modify (c_no char(5));
```

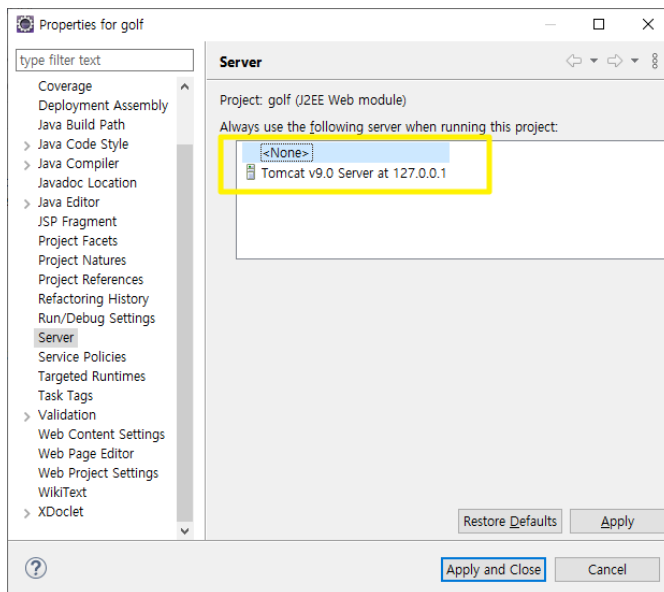
Table TBL_CLASS_202201이(가) 변경되었습니다.

Table TBL_CLASS_202201이(가) 변경되었습니다.

<eclipse 실행>

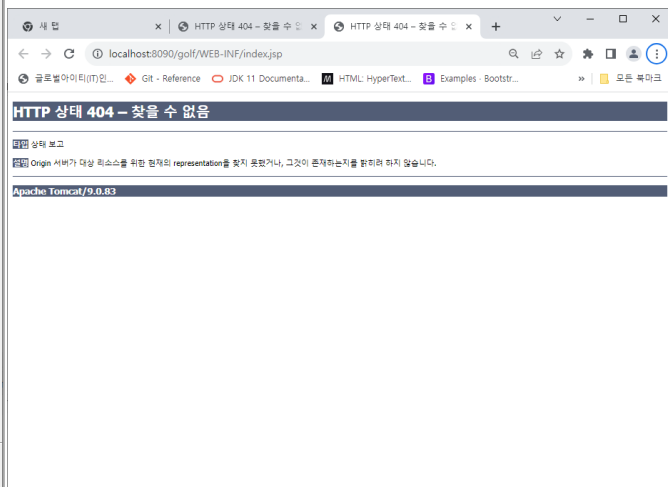
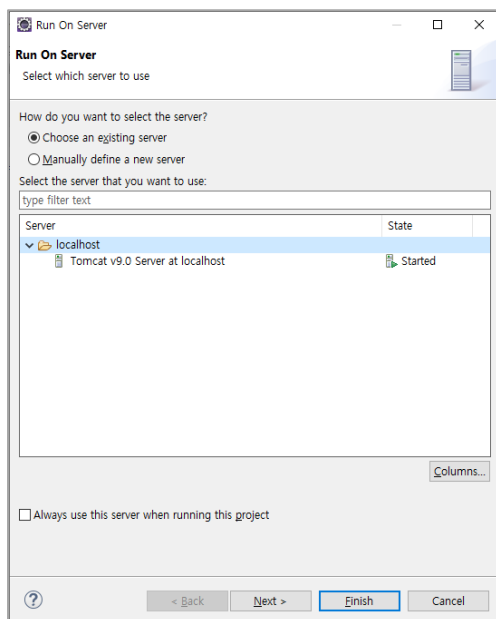
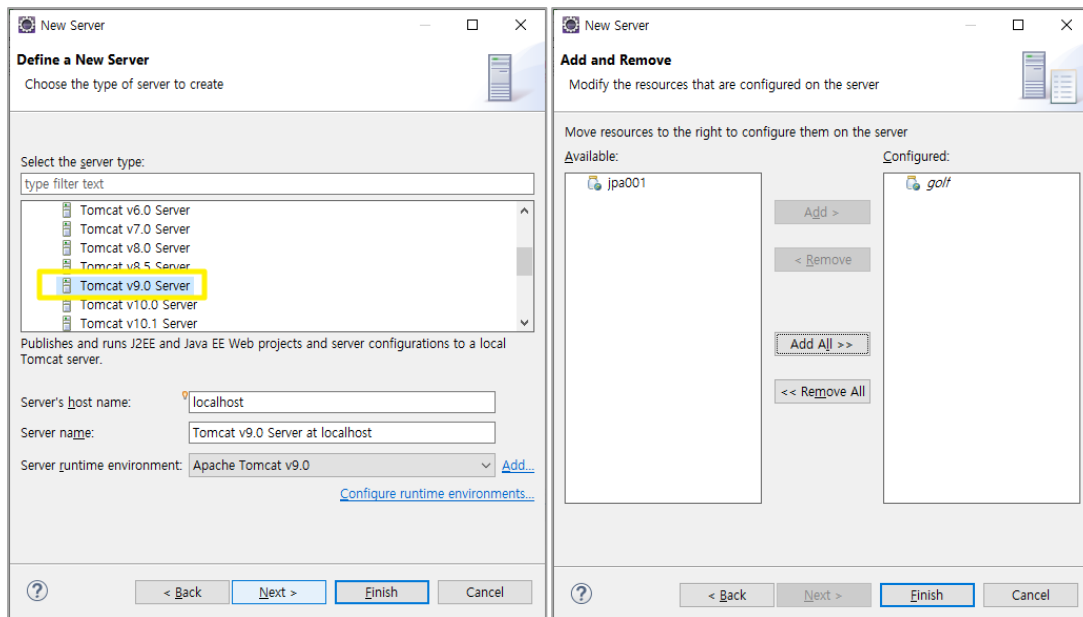


아파치 톰캣으로 target runtime을 설정해주고 시작 그러면 golf라는 project가 생성된다.



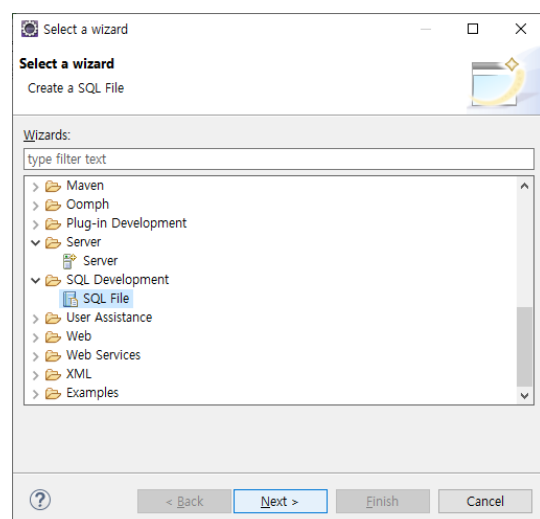
file > properties 에서 서버가 tomcat 으로 되어 있는지 확인!

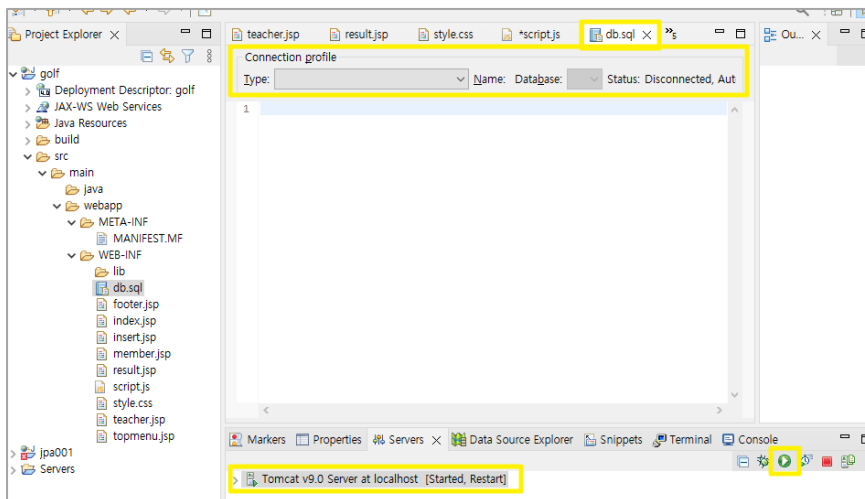
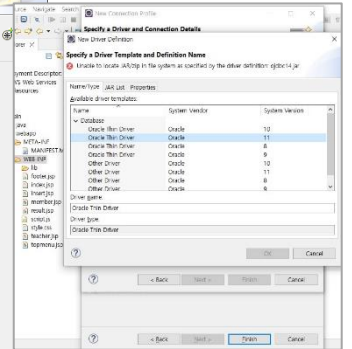
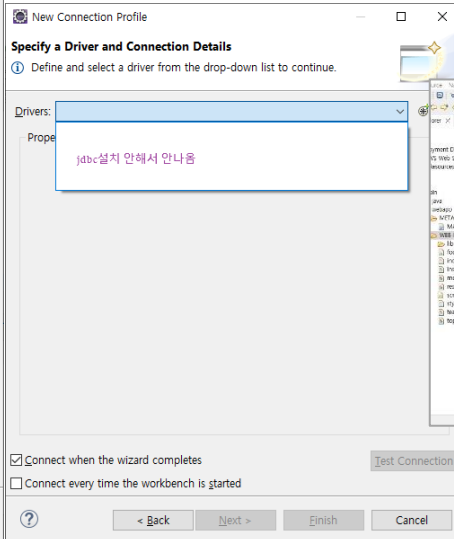
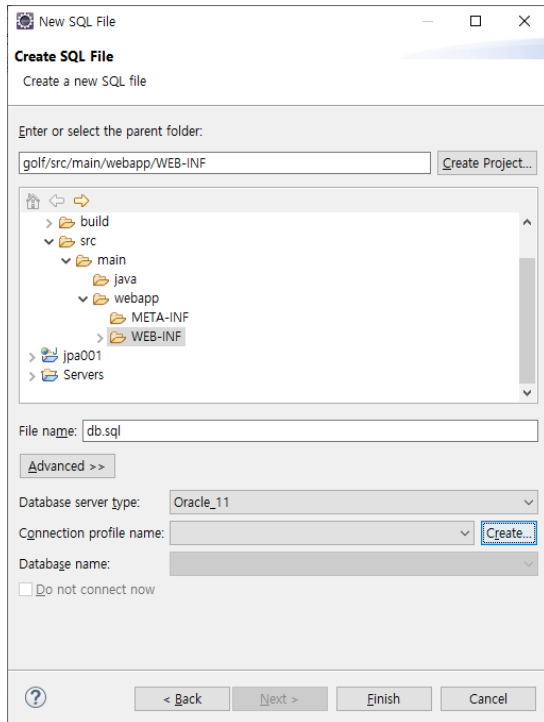
이후 Ctrl +f11 (run on server)에서 직접 브라우저로 연결할 수 있다. (톰캣과 연결)



연결할 수 없음을 확인했다

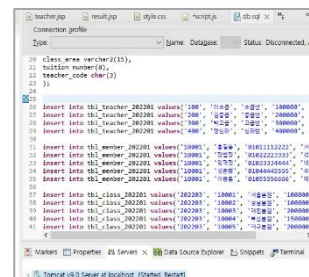
그 다음에 sQL file 과 연결하는 것을 확인했다.





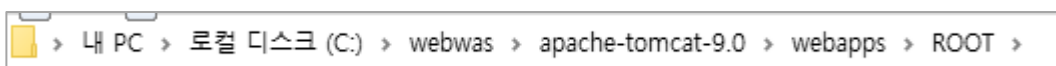
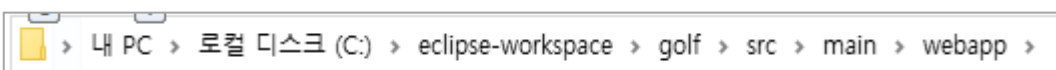
SQL 파일 실행했을 때 화면이다. JDBC 선택하지 않고 type 도 오라클 11 로 해놓지 않고, 빈칸으로 둔 채 실행시킨 화면이다. JDBC 는 자바언어 배우고 할 것.

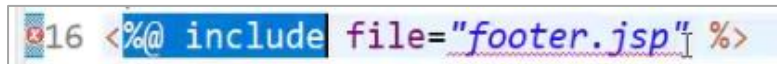
오라클 developer 에서 작업했던 것을 db.sql 에 붙여넣었다



이후 파일을 톰캣 ROOT 에 옮겨서 작동하는지 확인하는 작업을 했다

(파일을 옮기지 않으면 브라우저 127.0.0.1:8090 에서 작동이 되지 않음 -> 톰캣에서 작동되지 않아도 이클립스에서 바로 실행할 수 있음(ctrl+F11))



A screenshot of a code editor window. The editor shows a line of JSP code: `<%@ include file="footer.jsp" %>`. The text is color-coded: `<%@` is blue, `include` is blue, `file="` is black, `footer.jsp` is red and underlined, and `" %>` is black. A cursor is positioned at the end of the line, after the closing tag. The line number '16' is visible on the left side of the editor.

jsp 파일 자동으로 열고자 할 때 태그이다.

톰캣8090에서 보이는 것 보여주는 건 톰캣에서 하는 것
관리는 이클립스