

1) Oracle DB 생성 실습¹

- sqldeveloper 실행(스키마 생성)

새로 만들기/데이터베이스 접속 선택

접속 이름: system, 접속 세부정보: system@//loc...

Name: system, Color: [icon]

데이터베이스 유형: Oracle

사용자 정보 | 프록시 사용자

인증 유형: 기본값

사용자 이름(U): system, 룰(L): 기본값

비밀번호(P): [masked], ☐ 비밀번호 저장(V)

접속 유형(Y): 기본

세부정보 | 고급

호스트 이름(A): localhost

포트(B): 1521

☒ SID(I): xe

☐ 서비스 이름(E): [empty]

상태: 성공

도움말(H) | 저장(S) | 지우기(C) | 테스트(T) | 접속(O) | 취소

- Name 작성
- 사용자이름, 비밀번호 : system / 1234
- 테스트(상태:성공) > 저장 > 접속

- Oracle DB 작성시 주의사항 : **노란색 블러 표시**에서 하나씩 ctrl+Enter 를 쳐야 입력됨

순서	컬럼ID	컬럼명	형태	길이	NULL	비고
1	V_JUMIN	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_AREA	투표장소	CHAR	20		
6	V_CONFIRM	유권자확인	CHAR	1		

컬럼명은 미작성에 주의

- 소스 코드

```
create table tbl_vote_202005(
v_jumin char(13) not null primary key,
v_name varchar2(20),
m_no char(1),
v_time char(4),
v_area char(20),
v_confirm char(1))
```

¹ [\[정보처리산업기사 과정평가형\] 지역구의원투표 - DB 생성하기](#)

```

);

CREATE table tbl_member_202005(
m_no char(1) not null primary key,
m_name VARCHAR2(20),
p_code char(2),
p_school char(1),
m_jumin char(13),
m_city varchar2(20)
);

create table tbl_party_202005(
p_code char(2) not null primary key,
p_name varchar2(20),
p_indate DATE,
p_reader varchar2(20),
p_tel1 char(3),
p_tel2 char(4),
p_tel3 char(4)
);

select * from tbl_vote_202005;
select * from tbl_member_202005;
select * from tbl_party_202005;

insert into tbl_member_202005 values ('1', '김 후보', 'P1', '1', '6603011999991', '수선 화동');
insert into tbl_member_202005 values ('2', '이 후보', 'P2', '3', '5503011999992', '민들래동');
insert into tbl_member_202005 values ('3', '박 후보', 'P3', '2', '7703011999993', '나팔꽃동');
insert into tbl_member_202005 values ('4', '조 후보', 'P4', '2', '8803011999994', '진달래동');
insert into tbl_member_202005 values ('5', '최 후보', 'P5', '3', '9903011999995', '개나리동');

insert into tbl_party_202005 values ('P1', 'A정당', '2010-01-01', '위대표', '02', '1111', '0001');
insert into tbl_party_202005 values ('P2', 'B정당', '2010-02-01', '명대표', '02', '1111', '0002');
insert into tbl_party_202005 values ('P3', 'C정당', '2010-03-01', '기대표', '02', '1111', '0003');
insert into tbl_party_202005 values ('P4', 'D정당', '2010-04-01', '옥대표', '02', '1111', '0004');
insert into tbl_party_202005 values ('P5', 'E정당', '2010-05-01', '임대표', '02', '1111', '0005');

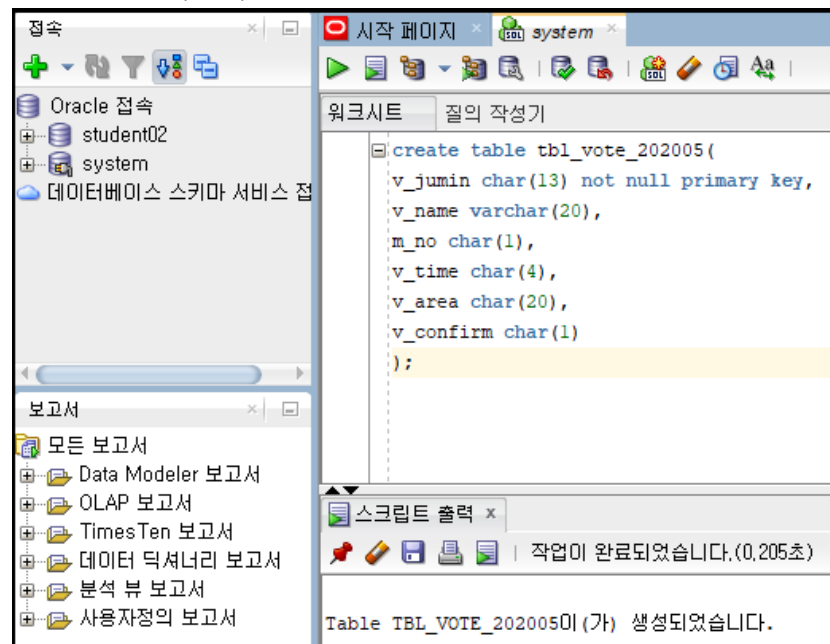
insert into tbl_vote_202005 values ('99010110001', '김유권', '1', '0930', '제1투표장', 'N');
insert into tbl_vote_202005 values ('89010120002', '이유권', '2', '0930', '제1투표장', 'N');
insert into tbl_vote_202005 values ('69010110003', '박유권', '3', '0930', '제1투표장', 'Y');
insert into tbl_vote_202005 values ('59010120004', '홍유권', '4', '0930', '제1투표장', 'Y');
insert into tbl_vote_202005 values ('79010110005', '조유권', '5', '0930', '제1투표장', 'Y');
insert into tbl_vote_202005 values ('89010120006', '최유권', '1', '0930', '제1투표장', 'Y');
insert into tbl_vote_202005 values ('59010110007', '지유권', '1', '0930', '제1투표장', 'Y');
insert into tbl_vote_202005 values ('49010120008', '장유권', '3', '0930', '제1투표장', 'Y');
insert into tbl_vote_202005 values ('79010110009', '정유권', '3', '0930', '제1투표장', 'Y');
insert into tbl_vote_202005 values ('89010120010', '강유권', '4', '0930', '제1투표장', 'Y');
insert into tbl_vote_202005 values ('99010110011', '신유권', '5', '0930', '제1투표장', 'Y');
insert into tbl_vote_202005 values ('79010120012', '오유권', '1', '1330', '제1투표장', 'Y');

insert into tbl_vote_202005 values ('69010110013', '현유권', '4', '1330', '제2투표장', 'Y');
insert into tbl_vote_202005 values ('89010110014', '왕유권', '2', '1330', '제2투표장', 'Y');
insert into tbl_vote_202005 values ('99010110015', '유유권', '3', '1330', '제2투표장', 'Y');
insert into tbl_vote_202005 values ('79010110016', '한유권', '2', '1330', '제2투표장', 'Y');
insert into tbl_vote_202005 values ('89010110017', '문유권', '4', '1330', '제2투표장', 'Y');
insert into tbl_vote_202005 values ('99010110018', '양유권', '2', '1330', '제2투표장', 'Y');
insert into tbl_vote_202005 values ('99010110019', '구유권', '4', '1330', '제2투표장', 'Y');
insert into tbl_vote_202005 values ('79010110020', '황유권', '5', '1330', '제2투표장', 'Y');
insert into tbl_vote_202005 values ('69010110021', '배유권', '3', '1330', '제2투표장', 'Y');
insert into tbl_vote_202005 values ('79010110022', '전유권', '3', '1330', '제2투표장', 'Y');
insert into tbl_vote_202005 values ('99010110023', '고유권', '1', '1330', '제2투표장', 'Y');
insert into tbl_vote_202005 values ('59010110024', '권유권', '3', '1330', '제2투표장', 'Y');

```

1-1) 테이블 생성

- Create table 테이블명(내용);



); 에서 **ctrl+Enter** 해야 생성된다.

- 좌측 + 로 새로운 db 생성
- **ctrl + Enter** : 명령어 입력
- 생성 확인
- **varchar2** 는 4kbyte 라서 무한정을 의미

1-2) 테이블 확인

- select * from all_tables;

	OWNER	TABLE_NAME	TABLESPACE_NAME	CLUSTER_NAME	IOT_NAME	STATUS	PCT_FREE	PCT_USED
1	SYS	ICOL\$	SYSTEM	C_OBJ#	(null)	VALID	0	
2	SYS	CON\$	SYSTEM	(null)	(null)	VALID	10	4
3	SYS	UNDO\$	SYSTEM	(null)	(null)	VALID	10	4
4	SYS	PROXY_ROLE_DATA\$	SYSTEM	(null)	(null)	VALID	10	4
5	SYS	FILE\$	SYSTEM	(null)	(null)	VALID	10	4

보유한 모든 테이블이 확인됨

- select * from tbl_vote_202005;

	V_JUMIN	V_NAME	M_NO	V_TIME	V_AREA	V_CONF...

해당 테이블이 확인됨, 내용 미작성 상태

1-3) 테이블 내용 작성

- insert into tbl_party_202005 values('p1', 'A정당','2010-01-01','위대표','02','1111','0001')

정당코드	명칭	등록연월일	대표자	전화번호1	전화번호2	전화번호3
P1	A정당	2010-01-01	위대표	02	1111	0001
P2	B정당	2010-02-01	명대표	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

```
insert into tbl_party_202005 values('p1', 'A정당','2010-01-01','위대표','02','1111','0001');
insert into tbl_party_202005 values('p2', 'B정당','2010-02-01','명대표','02','1111','0002');
insert into tbl_party_202005 values('p3', 'C정당','2010-03-01','기대표','02','1111','0003');
insert into tbl_party_202005 values('p4', 'D정당','2010-04-01','옥대표','02','1111','0004');
insert into tbl_party_202005 values('p5', 'E정당','2010-05-01','임대표','02','1111','0005');
```

1 행 이 (가) 삽입되었습니다.

한줄씩 입력해야 하는 불편함

1-4) 에러 발생

```
p_indate DATE,
p_reader varchar(20),
p_tel1 char(3),
p_tel2 char(4),
p_tel3 char(4)
);

insert into tbl_party_202005 values('p1', 'A정당','2010-01-01','위대표','02','1111','0001');
insert into tbl_party_202005 values('p2', 'B정당','2010-02-01','명대표','02','1111','0002');
insert into tbl_party_202005 values('p3', 'C정당','2010-03-01','기대표','02','1111','0003');
insert into tbl_party_202005 values('p4', 'D정당','2010-04-01','옥대표','02','1111','0004');
insert into tbl_party_202005 values('p5', 'E정당','2010-05-01','임대표','02','1111','0005');

insert into tbl_member_202005 values ('1', '김후보', 'P1', '1', '6603011999991', '수선화동');
insert into tbl_member_202005 values ('2', '이후보', 'P2', '3', '5503011999992', '민들레동');
insert into tbl_member_202005 values ('3', '박후보', 'P3', '2', '7703011999993', '나팔꽃동');
insert into tbl_member_202005 values ('4', '조후보', 'P4', '2', '8803011999994', '진달래동');
insert into tbl_member_202005 values ('5', '최후보', 'P5', '3', '9903011999995', '개나리동');

insert into tbl_vote_202005 values ('99010110001', '김유권', '1', '0930', '제1투표장', 'N');
insert into tbl_vote_202005 values ('89010120002', '이유권', '2', '0930', '제1투표장', 'N');
insert into tbl_vote_202005 values ('69010110003', '박유권', '3', '0930', '제1투표장', 'Y');
insert into tbl_vote_202005 values ('59010120004', '홍유권', '4', '0930', '제1투표장', 'Y');
insert into tbl_vote_202005 values ('79010110005', '조유권', '5', '0930', '제1투표장', 'Y');

select * from tbl_vote_202005;
select * from tbl_member_202005;
select * from tbl_party_202005;
```

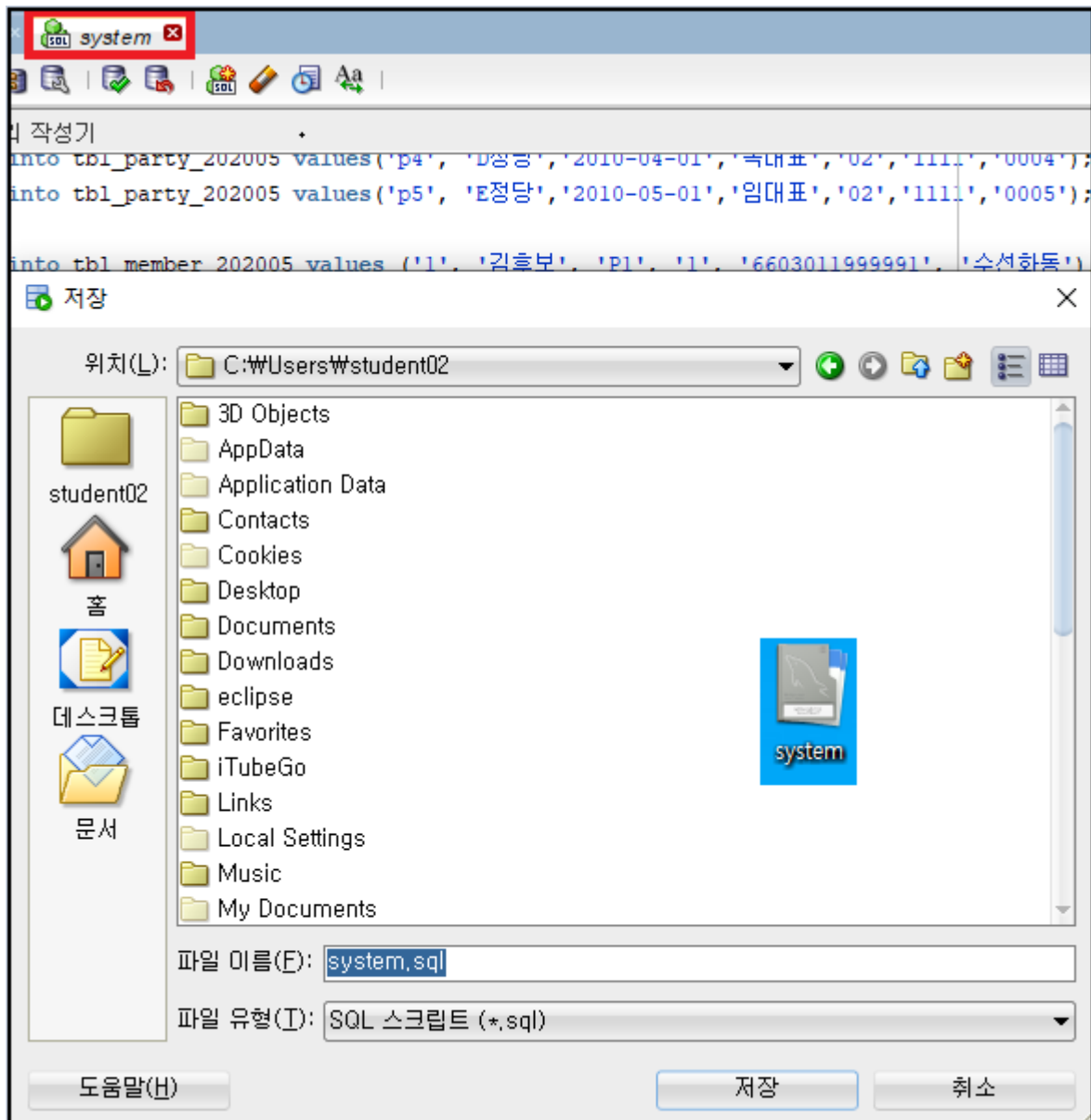
The screenshot shows the SQL Developer interface with a context menu open over the 'tbl_party_202005' table. The menu options include '접속(K)', '접속 해제(L)', '재접속(A)', '비밀번호 재설정(B)...', '스키마 브라우저(S)', 'DB 객체 찾기(F)', '전속 이력 바꾸기(C)', and '삭제(M)' (Delete). The '삭제(M)' option is highlighted with a red rectangle.

메모장 복붙 / 기존 db 삭제

- 오라클은 에러에 취약한 편
- 기존 작성 내용을 메모장에 저장해 새로 생성이 권장
- 개발툴에서의 삭제 : 기존 db 삭제 및 local data도 삭제해도 오라클의 특징상 잔존함
 - local data 경로 : C:\Users\student02\AppData\Roaming\SQL Developer
- 오라클의 특징 : 메모리에는 여전히 상주중이므로 에러가 나옴
 - mariadb 와 mysql 은 drop 시 사라지나 oracle은 남아있는 편
 - 정부기관이나 은행용이므로 오라클자체를 삭제해야 삭제됨
 - 즉, 오라클 삭제 전까진 drop 해도 데이터가 유지됨

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

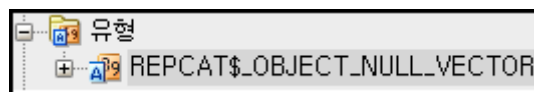
1-5) 저장



- developer 가 한글 미번역, eu-kr로 저장하므로, VS Code에서 파일 로드시 한글 깨짐

1-6) 삭제

- 초기화 : [Oracle 데이터베이스에서 전체 스키마 삭제](#)
- 유형 삭제 : run sql 및 delete 가 안먹을때 유형을 삭제



- 테이블 삭제시 : drop table 테이블명;

1-7) Table 및 소스 코드

- Table 상황

select * from DEPARTMENT;	
D_CODE	D_NAME
1 com	컴퓨터
2 han	국어
3 eng	영어

select * from STUDENT;				
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

select * from GRADE;		
GRADE	MIN	MAX
1 A	90	100
2 B	80	89
3 C	60	79
4 D	0	59

- 소스 코드

```
CREATE TABLE DEPARTMENT(
D_CODE VARCHAR2(10) NOT NULL,
D_NAME CHAR(30) NOT NULL,
CONSTRAINT PK_department PRIMARY KEY(D_CODE));
```

```
COMMENT ON TABLE DEPARTMENT IS '학과';
COMMENT ON COLUMN DEPARTMENT.D_CODE IS '학과코드';
COMMENT ON COLUMN DEPARTMENT.D_NAME IS '학과명';
```

```
CREATE TABLE GRADE (
GRADE VARCHAR2(5) NOT NULL,
MIN NUMBER NOT NULL,
MAX NUMBER NOT NULL,
CONSTRAINT PK_GRADE PRIMARY KEY(GRADE));
```

```
COMMENT ON TABLE GRADE IS '성적 등급';
COMMENT ON COLUMN GRADE.GRADE IS '등급';
COMMENT ON COLUMN GRADE.MIN IS '최저값';
COMMENT ON COLUMN GRADE.MAX IS '최대값';
```

```
CREATE TABLE STUDENT(S_NO NUMBER NOT NULL,
S_NAME VARCHAR2(10),
```

```

D_CODE VARCHAR2(10),
S_SENIOR VARCHAR2(10),
S_GRADE NUMBER,
CONSTRAINT PK_STUDENT PRIMARY KEY(S_NO),
CONSTRAINT FK_STUDENT FOREIGN KEY(D_CODE) REFERENCES DEPARTMENT(D_CODE));

```

```

COMMENT ON TABLE STUDENT IS '학생 테이블';
COMMENT ON COLUMN STUDENT.S_NO IS '학번';
COMMENT ON COLUMN STUDENT.S_NAME IS '이름';
COMMENT ON COLUMN STUDENT.D_CODE IS '학과코드(FK)';
COMMENT ON COLUMN STUDENT.S_SENIOR IS '선배';
COMMENT ON COLUMN STUDENT.S_GRADE IS '성적';

```

```

INSERT INTO DEPARTMENT(D_CODE, D_NAME) VALUES('com', '컴퓨터');
INSERT INTO DEPARTMENT(D_CODE, D_NAME) VALUES('han', '국어');
INSERT INTO DEPARTMENT(D_CODE, D_NAME) VALUES('eng', '영어');

```

```

INSERT INTO GRADE(GRADE, MIN, MAX) VALUES('A', '90', '100');
INSERT INTO GRADE(GRADE, MIN, MAX) VALUES('B', '80', '89');
INSERT INTO GRADE(GRADE, MIN, MAX) VALUES('C', '60', '79');
INSERT INTO GRADE(GRADE, MIN, MAX) VALUES('D', '0', '59');

```

```

INSERT INTO STUDENT(S_NO, S_NAME, D_CODE, S_SENIOR, S_GRADE) VALUES('15', '김가람', 'com', null, '83');
INSERT INTO STUDENT(S_NO, S_NAME, D_CODE, S_SENIOR, S_GRADE) VALUES('16', '정다솜', 'han', null, '96');
INSERT INTO STUDENT(S_NO, S_NAME, D_CODE, S_SENIOR, S_GRADE) VALUES('17', '이진수', 'com', '15', '95');
INSERT INTO STUDENT(S_NO, S_NAME, D_CODE, S_SENIOR, S_GRADE) VALUES('19', '황명훈', 'han', '16', '75');
INSERT INTO STUDENT(S_NO, S_NAME, D_CODE, S_SENIOR, S_GRADE) VALUES('37', '김아름', null, '17', '55');

```

```

select * from DEPARTMENT;
select * from GRADE;
select * from STUDENT;

```

```

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;

```

```

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
order by s.S_NO asc, s.S_Name desc;

```

```

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
order by s.S_NO asc, s.S_Name desc;

```

```

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
order by s.S_NO asc, s.S_Name desc;

```

```

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

```

```

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;

```

1-8) join : 2개 이상의 테이블을 연결하여 검색, pk 와 fk 값을 이용하여 검색

- Equi Join (inner join) : 일치하는 값, 둘다 존재하는 값을 조인

```
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;
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;
```

(위 아래 sql 구문은 동일)

```
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;
```

```
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;
```

- s, d 가 각 테이블을 의미

D_CODE	D_NAME
1 com	컴퓨터
2 han	국어
3 eng	영어

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



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

- join 시 참조되는 데이터의 null 값인 경우 null 로 나옴 : **s.S_SENIOR** 추가됨

```
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;
```

```
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;
```

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

1-9) outer join

- full outer join : 두개의 테이블을 다 합쳐서 호출

```
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
order by s.S_NO asc, s.S_Name desc;
```

	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

1-10) left & right outer join

- From STUDENT (왼쪽) s left outer join DEPARTMENT d

```
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
order by s.S_NO asc, s.S_Name desc;
```

	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)

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

	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

•

SELECT s.S_NO, s.S_NAME, s.D_CODE, d.D_NAME FROM STUDENT s right outer join DEPARTMENT d on s.D_CODE= d.D_CODE;	1	15 김가람	com	컴퓨터
	2	16 정다솜	han	국어
	3	17 이진수	com	컴퓨터
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;	4	19 황명훈	han	국어
	5	(null) (null)	(null)	영어

- 2) 예시 2
- 테이블 상황

2) 데이터 입출력 요건에 맞게 감사정보, 회원정보, 수강정보 테이블을 생성하시오.

가) 감사정보 (테이블 명 : TBL_TEACHER_202201)

순서	컬럼ID	컬럼명	형태	길이	NULL	비고
1	TEACHER_CODE	강사코드	CHAR	3	NOT NULL	PRIMARY KEY
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	정심화	심화반	400000	20220104

나) 회원정보 (테이블 명 : TBL_MEMBER_202201)

순서	컬럼ID	컬럼명	형태	길이	NULL	비고
1	C_NO	회원번호	CHAR	5	NOT NULL	PRIMARY KEY
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	장봉장	01022223333	성남시 분당구	일반
10003	임옥경	01033334444	대전시 유성구	일반
20001	성준형	01044445555	부산시 사구	VIP
20002	이봉룡	01055556666	대구시 북구	VIP

다) 수강정보 (테이블 명 : TBL_CLASS_202201)

순서	컬럼ID	컬럼명	형태	길이	NULL	비고
1	REGIST_MONTH	수강월	VARCHAR2	6	NOT NULL	PRIMARY KEY
2	C_NO	회원번호	CHAR	5	NOT NULL	PRIMARY KEY
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	서울분원	100000	100
202203	10002	상남분원	100000	100
202203	10003	대천분원	200000	200
202203	20001	부산분원	150000	300
202203	20002	대구분원	200000	400

- 소스 코드