

23.11.27

🕒 작성 일시	@2023년 11월 27일 오전 11:03
🏷 태그	

VARCHAR2 4000BYTE 까지 가능한 코드

TRUNCATE TABLE 테이블 명 : 빈 테이블로 만든다.

oracle 작업 이미지

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_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		

[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	정유권	3	0930	제1투표장	Y
79010110009	황유권	3	0930	제1투표장	Y
89010120010	강유권	4	0930	제1투표장	Y
99010110011	신유권	5	0930	제1투표장	Y
79010120012	오유권	1	1330	제1투표장	Y
69010110013	원유권	4	1330	제2투표장	Y
89010110014	왕유권	2	1330	제2투표장	Y
99010110015	유유권	3	1330	제2투표장	Y
79010110016	한유권	2	1330	제2투표장	Y
89010110017	문유권	4	1330	제2투표장	Y
99010110018	양유권	2	1330	제2투표장	Y
99010110019	구유권	4	1330	제2투표장	Y
79010110020	백유권	5	1330	제2투표장	Y
69010110021	배유권	3	1330	제2투표장	Y
79010110022	전유권	3	1330	제2투표장	Y
99010110023	고유권	1	1330	제2투표장	Y
59010110024	연유권	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_JUMIN	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	전달래동
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	임대표	02	1111	0001
P2	B정당	2010-02-01	임대표	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 색상, 폰트 등 스타일 구성요소는 선택 사항이다.

ORACLE 작업 내용

create table tbl_vote_202005(

V_JUMIN CHAR(13) NOT NULL PRIMARY KEY,

V_NAME VARCHAR(20),

M_NO CHAR(1),

V_TIME CHAR(4),

V_AREA CHAR(20),

```

V_CONFIRM CHAR(1)
);

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)
);

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)
);

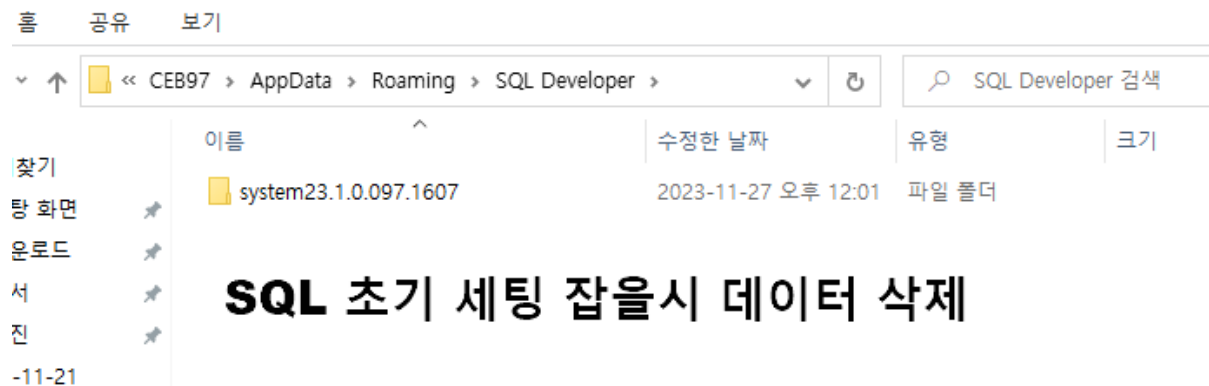
SELECT * FROM tbl_PARTY_202005;
SELECT * FROM TBL_MEMBER_202005;

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', '2',
'5503011999992', '민들래동');
INSERT INTO TBL_MEMBER_202005 VALUES ('3', '박후보', 'P3', '3',
'7703011999993', '나팔꽃동');

```

```
INSERT INTO TBL_MEMBER_202005 VALUES ('4', '조후보', 'P4', '4',
'88030119999994', '진달래동');
INSERT INTO TBL_MEMBER_202005 VALUES ('5', '최후보', 'P5', '5',
'99030119999995', '개나리동');
```



oracle은 한글 표기가 깨져서 나온다.(UTF-8 x)

```
INSERT INTO tbl_PARTY_202005 VALUES ('P1', 'A', '2010-01-01', 'g', '02', '1111', '0001');
INSERT INTO tbl_PARTY_202005 VALUES ('P2', 'B', '2010-02-01', 'g', '02', '1111', '0002');
INSERT INTO tbl_PARTY_202005 VALUES ('P3', 'C', '2010-03-01', 'g', '02', '1111', '0003');
INSERT INTO tbl_PARTY_202005 VALUES ('P4', 'D', '2010-04-01', 'g', '02', '1111', '0004');
INSERT INTO tbl_PARTY_202005 VALUES ('P5', 'E', '2010-05-01', 'g', '02', '1111', '0005');

INSERT INTO TBL_MEMBER_202005 VALUES ('1', 'I', 'P1', '1', '6603011999991', '0000');
INSERT INTO TBL_MEMBER_202005 VALUES ('2', 'I', 'P2', '2', '5503011999992', '0000');
INSERT INTO TBL_MEMBER_202005 VALUES ('3', 'I', 'P3', '3', '7703011999993', '0000');
INSERT INTO TBL_MEMBER_202005 VALUES ('4', 'I', 'P4', '4', '8803011999994', '0000');
INSERT INTO TBL_MEMBER_202005 VALUES ('5', 'I', 'P5', '5', '9903011999995', '0000');
```

오류 보고 -

ORA-00955: name is already used by an existing object

00955. 00000 - "name is already used by an existing object"

*Cause:

*Action: 삭제 했다는 기록도 로그에 남는다.

oracle에서는 외부 경로에서 테이블을 삭제 해도 데이터가 남는다.

	M_NO	M_NAME	P_CODE	P_SCHOOL	M_JUMIN	M_CITY
1	1	김후보	P1	1	6603011999991	수선화동
2	2	이후보	P2	2	5503011999992	민들래동
3	3	박후보	P3	3	7703011999993	나팔꽃동
4	4	조후보	P4	4	8803011999994	진달래동
5	5	최후보	P5	5	9903011999995	개나리동

ibm 공식 문서 URL : <https://www.ibm.com/docs/ko/openpages/8.3.0?topic=creation-dropping-full-schema-in-oracle-database>

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Object Explorer' pane displays a tree structure of the database objects, including a table named 'REPCAT\$.OBJECT_NULL_VECTOR'. The main window shows a SQL script for creating or replacing a type named 'repcat\$.object_null_vector AS OBJECT'. The script includes comments and defines attributes: 'type_owner' (VARCHAR2(30)), 'type_name' (VARCHAR2(30)), 'type_hashcode' (RAW(17)), and 'null_vector' (RAW(2000)).

```
create or replace TYPE repcat$.object_null_vector AS OBJECT
(
  -- type owner, name, hashcode for the type represented by null_vector
  type_owner  VARCHAR2(30),
  type_name   VARCHAR2(30),
  type_hashcode RAW(17),
  -- null_vector for a particular object instance
  -- ROBJ REVISIT: should only contain the null image, and not version#
  null_vector RAW(2000)
)
```

삭제 명령어 적용 후에도 명령어가 적용 안될지 해당 페이지 구문 삭제

JOIN

- 2개 이상의 테이블을 연결 검색

- PK와 FK 값을 이용하여 검색

s. 테이블명
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;

일치하는 값
 둘다 존재하는 값만 출력

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;

s의 있는 코드와 d에있는 코드가
 일치할때 값을 출력

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

D_CODE	D_NAME	GRADE	MIN	MAX
1 com	컴퓨터	1 A	90	100
2 han	국어	2 B	80	89
3 eng	영어	3 C	60	79
4 D		4 D	0	59

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

S_NO	S_NAME	D_CODE	D_NAME
1	김가람	com	컴퓨터
2	정다솜	han	국어
3	이진수	com	컴퓨터
4	황영훈	han	국어

ORDER BY

SELECT s.S_NO, 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;

asc : 오름 차순 desc : 내림 차순

S_NO	S_SENIOR	D_CODE	D_NAME
1	15 (null)	com	컴퓨터
2	16 (null)	han	국어
3	17 15	com	컴퓨터
4	19 16	han	국어

FULL OUTER JOIN

```
SELECT s.S_NO, s.S_NAME, 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;
```

full outer join은 전부 출력
null = null

EQUI join
일치 하는 값
존재 하는 값 을 가져온다.

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
5	37 김아름	(null)	(null)
6	(null) (null)	(null)	영어

RIGHT/LEFT

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

RIGHT

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
5	(null) (null)	(null)	영어

```
SELECT s.S_NO, s.S_NAME, 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;
```

LEFT

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
5	37 김아름	(null)	(null)

REPARTMENT 의 값을 다 출력

||

```
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;
```

=

STUDENT의 값을 다 출력

```
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;
```

ecrips

```
<%@ include file="footer.jsp" %>
```