

DDL

(Data Definition Language)

DDL이란?

데이터 정의 언어이다. 객체(Object)를 만들고(CREATE), 수정하고(ALTER), 삭제(DROP)하는 구문을 말한다.

오라클 객체의 종류

테이블(TABLE), 뷰(VIEW), 시퀀스(SEQUENCE), 인덱스(INDEX), 패키지(PACKAGE), 프로시저(PROCEDURAL), 함수(FUNCTION), 트리거(TRIGGER), 동의어(SYNONYM), 사용자(USER)가 있다.

CREATE - 테이블만들기

[표현식]

```
CREATE TABLE 테이블명 (컬럼명 자료형(크기), 컬럼명 자료형(크기), ...);
```

```
CREATE TABLE MEMBER (  
    MEMBER_ID VARCHAR2(20),  
    MEMBER_PWD VARCHAR2(20),  
    MEMBER_NAME VARCHAR2(20)  
);
```

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1	MEMBER_ID	VARCHAR2 (20 BYTE)	Yes	(null)	1 (null)	
2	MEMBER_PWD	VARCHAR2 (20 BYTE)	Yes	(null)	2 (null)	
3	MEMBER_NAME	VARCHAR2 (30 BYTE)	Yes	(null)	3 (null)	

오라클의 데이터형

데이터형	설명
CHAR(크기)	고정길이 문자 데이터
VARCHAR2(크기)	가변길이 문자 데이터(최대 2,000 Byte)
NUMBER	숫자 데이터(최대 40자리)
NUMBER(길이)	숫자 데이터로, 길이 지정 가능하다 (최대 38자리)
DATE	날짜 데이터(BC 4712년 1월 1일 ~ AD 4712년 12월 31일)
LONG	가변 길이 문자형 데이터(최대 2GB)
LOB	2GB까지의 가변길이 바이너리 데이터 저장 가능 (이미지, 실행파일 등을 저장할 수 있음)
ROWID	DB에 저장되지 않는 행을 식별할 수 있는 고유 값
BFILE	대용량의 바이너리 데이터 저장 가능(최대 4GB)
TIMESTAMP	DATE형의 확장된 형태이다.
INTERVAL YEAR TO MONTH	년과 월을 이용하여 기간을 저장한다.
INTERVAL DAY TO SECONT	일, 시, 분, 초를 이용하여 기간을 저장한다.

DDL(Data Definition Language) KH KH정보교육원

컬럼 주석

[표현식]

COMMENT ON COLUMN 테이블명.컬럼명 IS '주석내용';

COMMENT ON COLUMN MEMBER.MEMBER_ID IS '회원아이디';
COMMENT ON COLUMN MEMBER.MEMBER_PWD IS '비밀번호';
COMMENT ON COLUMN MEMBER.MEMBER_NAME IS '회원이름';

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	MEMBER_ID	VARCHAR2(20 BYTE)	Yes	(null)	1	회원아이디
2	MEMBER_PWD	VARCHAR2(20 BYTE)	Yes	(null)	2	비밀번호
3	MEMBER_NAME	VARCHAR2(30 BYTE)	Yes	(null)	3	회원이름

제약 조건(CONSTRAINTS)

테이블 작성시 각 컬럼에 대한 기록에 대해 제약조건을 설정할 수 있다.
데이터 무결성 보장이 주 목적으로 이다. 입력 데이터에 문제가 없는지
검사와 데이터의 수정/삭제 가능 여부 검사 등을 위해 사용한다.

제약 조건	설명
NOT NULL	데이터에 NULL을 허용하지 않는다.
UNIQUE	중복된 값을 허용하지 않는다.
PRIMARY KEY	NULL을 허용하지 않고, 중복 값을 허용하지 않는다. 컬럼의 고유 식별자로 사용하기 위함이다.
FOREIGN KEY	참조되는 테이블의 컬럼의 값이 존재하면 허용한다.
CHECK	저장 가능한 데이터 값의 범위나 조건을 지정하여 설정한 값만 허용한다.

DDL(Data Definition Language) KH KH정보교육원

제약 조건(CONSTRAINTS) 확인

```
DESC USER_CONSTRAINTS;
```

이름	널	유형
OWNER		VARCHAR2(120)
CONSTRAINT_NAME	NOT NULL	VARCHAR2(30)
CONSTRAINT_TYPE		VARCHAR2(1)
TABLE_NAME	NOT NULL	VARCHAR2(30)
SEARCH_CONDITION		LONG
R_OWNER		VARCHAR2(120)
R_CONSTRAINT_NAME		VARCHAR2(30)
DELETE_RULE		VARCHAR2(9)
STATUS		VARCHAR2(8)
DEFERRABLE		VARCHAR2(14)
DEFERRED		VARCHAR2(9)
VALIDATED		VARCHAR2(13)
GENERATED		VARCHAR2(14)
BAD		VARCHAR2(3)
RELY		VARCHAR2(4)
LAST_CHANGE		DATE
INDEX_OWNER		VARCHAR2(30)
INDEX_NAME		VARCHAR2(30)
INVALID		VARCHAR2(7)
VIEW_RELATED		VARCHAR2(14)

```
DESC USER_CONS_COLUMNS;
```

이름	널	유형
OWNER	NOT NULL	VARCHAR2(30)
CONSTRAINT_NAME	NOT NULL	VARCHAR2(30)
TABLE_NAME	NOT NULL	VARCHAR2(30)
COLUMN_NAME		VARCHAR2(4000)
POSITION		NUMBER

DDL(Data Definition Language) KH KH정보교육원

NOT NULL

해당 컬럼에 반드시 값이 기록되어야 하는 경우, 특정 컬럼에 값을 저장하거나 수정할 때 NULL 값을 허용하지 않도록 컬럼 레벨에서 제한한다.

```
CREATE TABLE USER_NOCONS(  
  USER_NO NUMBER,  
  USER_ID VARCHAR2(20),  
  USER_PWD VARCHAR2(30),  
  USER_NAME VARCHAR2(30),  
  GENDER VARCHAR2(10),  
  PHONE VARCHAR2(30),  
  EMAIL VARCHAR2(50)  
);
```

Table USER_NOCONS이 (가) 생성되었습니다.

```
INSERT INTO USER_NOCONS  
VALUES(1, 'user01', 'pass01', '홍길동', '남',  
      '010-1234-5678', 'hong123@kh.or.kr');
```

1 행 이 (가) 삽입되었습니다.

```
INSERT INTO USER_NOCONS  
VALUES(2, NULL, NULL, NULL, NULL,  
      '010-1234-5678', 'hong123@kh.or.kr');
```

1 행 이 (가) 삽입되었습니다.

	USER_NO	USER_ID	USER_PWD	USER_NAME	GENDER	PHONE	EMAIL
1	1	USER01	PASS01	홍길동	남	010-1234-5678	hong123@kh.or.kr
2	2	(null)	(null)	(null)	(null)	010-1234-5678	hong123@kh.or.kr

** 컬럼에 아무 제약조건을 설정하지 않을 경우, NULL값이 문제 없이 삽입됨

DDL(Data Definition Language) KH KH정보교육원

NOT NULL

```
CREATE TABLE USER_NOTNULL(  
  USER_NO NUMBER NOT NULL,  
  USER_ID VARCHAR2(20) NOT NULL,  
  USER_PWD VARCHAR2(30) NOT NULL,  
  USER_NAME VARCHAR2(30),  
  GENDER VARCHAR2(10),  
  PHONE VARCHAR2(30),  
  EMAIL VARCHAR2(50)  
);
```

Table USER_NOTNULL이 (가) 생성되었습니다.

```
INSERT INTO USER_NOTNULL  
VALUES(1, 'user01', 'pass01', '홍길동', '남',  
      '010-1234-5678', 'hong123@kh.or.kr');
```

1 행 이 (가) 삽입되었습니다.

```
INSERT INTO USER_NOTNULL  
VALUES(2, NULL, NULL, NULL, NULL,  
      '010-1234-5678', 'hong123@kh.or.kr');
```

오류 보고 -

SQL 오류: ORA-01400: cannot insert NULL into ("EMPLOYEE"."USER_NOTNULL"."USER_ID")
01400. 00000 - "cannot insert NULL into (%s)"

*Cause: An attempt was made to insert NULL into previously listed objects.

*Action: These objects cannot accept NULL values.

	USER_NO	USER_ID	USER_PWD	USER_NAME	GENDER	PHONE	EMAIL
1	1	user01	pass01	홍길동	남	010-1234-5678	hong123@kh.or.kr

** NOT NULL 제약조건이 설정된 컬럼에 NULL값이 입력되면, 행 자체를 삽입하지 않는다.

DDL(Data Definition Language) KH KH정보교육원

UNIQUE

컬럼에 입력 값에 대해 중복을 제한하는 제약조건이다. 컬럼레벨과 테이블레벨에 설정 가능하다.

```
CREATE TABLE USER_NOCONS(  
  USER_NO NUMBER,  
  USER_ID VARCHAR2(20),  
  USER_PWD VARCHAR2(30),  
  USER_NAME VARCHAR2(30),  
  GENDER VARCHAR2(10),  
  PHONE VARCHAR2(30),  
  EMAIL VARCHAR2(50)  
);
```

```
INSERT INTO USER_NOCONS  
VALUES(1, 'user01', 'pass01', NULL, NULL,  
      '010-1234-5678', 'hong123@kh.or.kr');
```

1 행 이 (가) 삽입되었습니다.

Table USER_NOCONS이 (가) 생성되었습니다.

	USER_NO	USER_ID	USER_PWD	USER_NAME	GENDER	PHONE	EMAIL
1	1	user01	pass01	홍길동	남	010-1234-5678	hong123@kh.or.kr
2	2	(null)	(null)	(null)	(null)	010-1234-5678	hong123@kh.or.kr
3	1	user01	pass01	홍길동	남	010-1234-5678	hong123@kh.or.kr

** UNIQUE 제약조건이 없는 컬럼은 중복된 값도 저장을 허용한다.

DDL(Data Definition Language) KH KH정보교육원

UNIQUE

```
CREATE TABLE USER_UNIQUE(  
  USER_NO NUMBER,  
  USER_ID VARCHAR2(20) UNIQUE,  
  USER_PWD VARCHAR2(30) NOT NULL,  
  USER_NAME VARCHAR2(30),  
  GENDER VARCHAR2(10),  
  PHONE VARCHAR2(30),  
  EMAIL VARCHAR2(50)  
);
```

Table USER_UNIQUE이 (가) 생성되었습니다.

```
INSERT INTO USER_UNIQUE  
VALUES(1, 'user01', 'pass01', '홍길동', '남',  
      '010-1234-5678', 'hong123@kh.or.kr');
```

1 행 이 (가) 삽입되었습니다.

```
INSERT INTO USER_UNIQUE  
VALUES(1, 'user01', 'pass01', NULL, NULL,  
      '010-1234-5678', 'hong123@kh.or.kr');
```

오류 보고 -

SQL 오류: ORA-00001: unique constraint (EMPLOYEE, SYS_C007182) violated
00001. 00000 - "unique constraint (%s.%s) violated"

*Cause: An UPDATE or INSERT statement attempted to insert a duplicate key.
For Trusted Oracle configured in DBMS MAC mode, you may see
this message if a duplicate entry exists at a different level.

*Action: Either remove the unique restriction or do not insert the key.

```
SELECT UCC.TABLE_NAME, UCC.COLUMN_NAME, UC.CONSTRAINT_TYPE  
FROM USER_CONSTRAINT UC, USER_CONS_COLUMNS UCC  
WHERE UCC.CONSTRAINT_NAME = UC.CONSTRAINT_NAME  
AND UCC.CONSTRAINT_NAME = 'SYS_C007182';
```

	TABLE_NAME	COLUMN_NAME	CONSTRAINT_TYPE
1	USER_UNIQUE	USER_ID	U

DDL(Data Definition Language) KH KH정보교육원

UNIQUE

```
CREATE TABLE USER_UNIQUE2(  
  USER_NO NUMBER,  
  USER_ID VARCHAR2(20),  
  USER_PWD VARCHAR2(30) NOT NULL,  
  USER_NAME VARCHAR2(30),  
  GENDER VARCHAR2(10),  
  PHONE VARCHAR2(30),  
  EMAIL VARCHAR2(50),  
  UNIQUE (USER_ID) -- 테이블레벨  
);
```

Table USER_UNIQUE2이(가) 생성되었습니다.

```
INSERT INTO USER_UNIQUE2  
VALUES(1, 'user01', 'pass01', '홍길동', '남',  
      '010-1234-5678', 'hong123@kh.or.kr');
```

1 행 이(가) 삽입되었습니다.

```
INSERT INTO USER_UNIQUE2  
VALUES(1, 'user01', 'pass01', NULL, NULL,  
      '010-1234-5678', 'hong123@kh.or.kr');
```

SQL 오류: ORA-00001: unique constraint (EMPLOYEE.SYS_C007184) violated
00001. 00000 - "unique constraint (%s.%s) violated"

*Cause: An UPDATE or INSERT statement attempted to insert a duplicate key.
For Trusted Oracle configured in DBMS MAC mode, you may see
this message if a duplicate entry exists at a different level.

*Action: Either remove the unique restriction or do not insert the key.

	USER_NO	USER_ID	USER_PWD	USER_NAME	GENDER	PHONE	EMAIL
1	1	user01	pass01	홍길동	남	010-1234-5678	hong123@kh.or.kr

** 중복값이 있는 경우 UNIQUE 제약조건에 의해 행이 삽입되지 않는다.

UNIQUE

```
INSERT INTO USER_UNIQUE2  
VALUES(1, 'user01', 'pass01', '홍길동', '남',  
      '010-1234-5678', 'hong123@kh.or.kr');
```

| 1 행 이 (가) 삽입되었습니다.

```
INSERT INTO USER_UNIQUE2  
VALUES(1, NULL, NULL, '홍길동', '남',  
      '010-1234-5678', 'hong123@kh.or.kr');
```

| 1 행 이 (가) 삽입되었습니다.

```
INSERT INTO USER_UNIQUE2  
VALUES(1, NULL, NULL, '홍길동', '남',  
      '010-1234-5678', 'hong123@kh.or.kr');
```

| 1 행 이 (가) 삽입되었습니다.

	USER_NO	USER_ID	USER_PWD	USER_NAME	GENDER	PHONE	EMAIL
1	1	user01	pass01	홍길동	남	010-1234-5678	hong123@kh.or.kr
2	1	(null)	pass01	홍길동	남	010-1234-5678	hong123@kh.or.kr
3	1	(null)	pass01	홍길동	남	010-1234-5678	hong123@kh.or.kr

** UNIQUE 제약조건은 NULL값 중복 저장이 가능하다.

** 해결방법은 테이블 생성시 컬럼레벨에 NOT NULL을 함께 지정하면 된다.

UNIQUE

```
CREATE TABLE USER_UNIQUE3(  
  USER_NO NUMBER,  
  USER_ID VARCHAR2(20),  
  USER_PWD VARCHAR2(30) NOT NULL,  
  USER_NAME VARCHAR2(30),  
  GENDER VARCHAR2(10),  
  PHONE VARCHAR2(30),  
  EMAIL VARCHAR2(50),  
  UNIQUE (USER_NO, USER_ID)  
);
```

Table USER_UNIQUE3이 (가) 생성되었습니다.

TABLE_NAME	COLUMN_NAME	CONSTRAINT_NAME	CONSTRAINT_TYPE
USER_UNIQUE3	USER_NO	SYS_C007186	U
USER_UNIQUE3	USER_ID	SYS_C007186	U

** 두 개의 컬럼을 묶어서
하나의 UNIQUE 제약조건을 설정한다.

```
INSERT INTO USER_UNIQUE3  
VALUES(1, 'user01', 'pass01', '홍길동', '남',  
      '010-1234-5678', 'hong123@kh.or.kr');
```

1 행 이 (가) 삽입되었습니다.

```
INSERT INTO USER_UNIQUE3  
VALUES(2, 'user01', 'pass01', NULL, NULL,  
      '010-1234-5678', 'hong123@kh.or.kr');
```

1 행 이 (가) 삽입되었습니다.

```
INSERT INTO USER_UNIQUE3  
VALUES(2, 'user02', 'pass02', NULL, NULL,  
      '010-1234-5678', 'hong123@kh.or.kr');
```

1 행 이 (가) 삽입되었습니다.

```
INSERT INTO USER_UNIQUE3  
VALUES(1, 'user01', 'pass01', NULL, NULL,  
      '010-1234-5678', 'hong123@kh.or.kr');
```

SQL 오류: ORA-00001: unique constraint (EMPLOYEE.SYS C007186) violated
00001. 00000 - "unique constraint (%s.%s) violated"

*Cause: An UPDATE or INSERT statement attempted to insert a duplicate key.
For Trusted Oracle configured in DBMS MAC mode, you may see
this message if a duplicate entry exists at a different level.

*Action: Either remove the unique restriction or do not insert the key.

PRIMARY KEY

테이블에서 한 행의 정보를 구분하기 위한 고유 식별자(Identifier) 역할을 한다. NOT NULL의 의미와 UNIQUE의 의미를 둘 다 가지고 있으며, 한 테이블당 한 개만 설정할 수 있다.

컬럼 레벨과 테이블 레벨에서 둘 다 지정 가능하다.

DDL(Data Definition Language) KH KH정보교육원

PRIMARY KEY

```
CREATE TABLE USER_PRIMARYKEY(  
  USER_NO NUMBER PRIMARY KEY,  
  USER_ID VARCHAR2(20) UNIQUE,  
  USER_PWD VARCHAR2(30) NOT NULL,  
  USER_NAME VARCHAR2(30),  
  GENDER VARCHAR2(10),  
  PHONE VARCHAR2(30),  
  EMAIL VARCHAR2(50)  
);
```

혹은

```
CREATE TABLE USER_PRIMARYKEY(  
  USER_NO NUMBER,  
  USER_ID VARCHAR2(20) UNIQUE,  
  USER_PWD VARCHAR2(30) NOT NULL,  
  USER_NAME VARCHAR2(30),  
  GENDER VARCHAR2(10),  
  PHONE VARCHAR2(30),  
  EMAIL VARCHAR2(50),  
  PRIMARY KEY (USER_NO)  
);
```

Table USER_PRIMARYKEY이 (가) 생성되었습니다.

```
INSERT INTO USER_PRIMARYKEY  
VALUES(1, 'user01', 'pass01', '홍길동', '남',  
      '010-1234-5678', 'hong123@kh.or.kr');
```

1 행 이 (가) 삽입되었습니다.

```
INSERT INTO USER_PRIMARYKEY  
VALUES(1, 'user02', 'pass02', '이순신', '남',  
      '010-5678-9012', 'lee123@kh.or.kr');
```

SQL 오류: ORA-00001: unique constraint (EMPLOYEE.SYS_C007188) violated
00001. 00000 - "unique constraint (%s.%s) violated"

*Cause: An UPDATE or INSERT statement attempted to insert a duplicate key.
For Trusted Oracle configured in DBMS MAC mode, you may see
this message if a duplicate entry exists at a different level.

*Action: Either remove the unique restriction or do not insert the key.

```
INSERT INTO USER_PRIMARYKEY  
VALUES(NULL, 'user03', 'pass03', '유관순', '여',  
      '010-9999-3131', 'yoo123@kh.or.kr');
```

오류 보고 -

SQL 오류: ORA-01400: cannot insert NULL into ("EMPLOYEE"."USER_PRIMARYKEY"."USER_NO")
01400. 00000 - "cannot insert NULL into (%s)"

*Cause: An attempt was made to insert NULL into previously listed objects.

*Action: These objects cannot accept NULL values.

TABLE_NAME	COLUMN_NAME	CONSTRAINT_NAME	CONSTRAINT_TYPE
1 USER_PRIMARYKEY	USER_NO	SYS_C007188	P

DDL(Data Definition Language) KH KH정보교육원

PRIMARY KEY

```
CREATE TABLE USER_PRIMARYKEY2(  
  USER_NO NUMBER,  
  USER_ID VARCHAR2(20),  
  USER_PWD VARCHAR2(30) NOT NULL,  
  USER_NAME VARCHAR2(30),  
  GENDER VARCHAR2(10),  
  PHONE VARCHAR2(30),  
  EMAIL VARCHAR2(50),  
  PRIMARY KEY (USER_NO, USER_ID)  
);
```

Table USER_PRIMARYKEY2이 (가) 생성되었습니다.

TABLE_NAME	COLUMN_NAME	CONSTRAINT_NAME	CONSTRAINT_TYPE
1 USER_PRIMARYKEY2	USER_NO	SYS_C007196	P
2 USER_PRIMARYKEY2	USER_ID	SYS_C007196	P

★★ 두 개의 컬럼을 묶어서
하나의 PRIMARY KEY 제약조건을 설정한다.

	USER_NO	USER_ID	USER_PWD	USER_NAME	GENDER	PHONE	EMAIL
1	1	user01	pass01	홍길동	남	010-1234-5678	hong123@kh.or.kr
2	1	user02	pass02	이순신	남	010-5678-9012	lee123@kh.or.kr
3	2	user01	pass01	유관순	여	010-9999-3131	yoo123@kh.or.kr

```
INSERT INTO USER_PRIMARYKEY2  
VALUES(1, 'user01', 'pass01', '홍길동', '남',  
      '010-1234-5678', 'hong123@kh.or.kr');
```

1 행 이 (가) 삽입되었습니다.

```
INSERT INTO USER_PRIMARYKEY2  
VALUES(1, 'user02', 'pass02', '이순신', '남',  
      '010-5678-9012', 'lee123@kh.or.kr');
```

1 행 이 (가) 삽입되었습니다.

```
INSERT INTO USER_PRIMARYKEY2  
VALUES(2, 'user01', 'pass01', '유관순', '여',  
      '010-9999-3131', 'yoo123@kh.or.kr');
```

1 행 이 (가) 삽입되었습니다.

```
INSERT INTO USER_PRIMARYKEY2  
VALUES(1, 'user01', 'pass01', '신사임당', '여',  
      '010-9999-9999', 'sin123@kh.or.kr');
```

오류 보고 -

SQL 오류: ORA-00001: unique constraint (EMPLOYEE.SYS_C007196) violated
00001. 00000 - "unique constraint (%s.%s) violated"

*Cause: An UPDATE or INSERT statement attempted to insert a duplicate key.
For Trusted Oracle configured in DBMS MAC mode, you may see
this message if a duplicate entry exists at a different level.

*Action: Either remove the unique restriction or do not insert the key.

FOREIGN KEY

참조 무결성을 유지하기 위한 제약조건이다. 즉, 참조된 다른 테이블이 제공하는 값만 사용할 수 있도록 제한을 거는 것이다.

참조되는 컬럼과 참조된 컬럼을 통해 테이블간의 관계가 형성된다.

또한, 참조되는 값은 제공되는 값 이외에 NULL을 사용 가능하며, 참조할 테이블의 참조할 컬럼명을 생략할 경우에는 PRIMARY KEY로 설정된 컬럼이 자동으로 참조할 컬럼이 된다.

FOREIGN KEY

```
CREATE TABLE USER_GRADE(  
    GRADE_CODE NUMBER PRIMARY KEY  
    GRADE_NAME VARCHAR2(30) NOT NULL  
);
```

Table USER_GRADE이 (가) 생성되었습니다.

```
INSERT INTO USER_GRADE  
VALUES(10, '일반회원');
```

1 행 이 (가) 삽입되었습니다.

```
INSERT INTO USER_GRADE  
VALUES(20, '우수회원');
```

1 행 이 (가) 삽입되었습니다.

```
INSERT INTO USER_GRADE  
VALUES(30, '특별회원');
```

1 행 이 (가) 삽입되었습니다.

```
SELECT * FROM USER_GRADE;
```

	GRADE_...	GRADE_NAME
1	10	일반회원
2	20	우수회원
3	30	특별회원

FOREIGN KEY

```
CREATE TABLE USER_FOREIGNKEY(  
    USER_NO NUMBER PRIMARY KEY,  
    USER_ID VARCHAR2(20) UNIQUE,  
    USER_PWD VARCHAR2(30) NOT NULL,  
    USER_NAME VARCHAR2(30),  
    GENDER VARCHAR2(10),  
    PHONE VARCHAR2(30),  
    EMAIL VARCHAR2(50),  
    GRADE_CODE NUMBER,  
    FOREIGN KEY (GRADE_CODE) REFERENCES USER_GRADE (GRADE_CODE)  
);
```

혹은

```
CREATE TABLE USER_FOREIGNKEY(  
    USER_NO NUMBER PRIMARY KEY,  
    USER_ID VARCHAR2(20) UNIQUE,  
    USER_PWD VARCHAR2(30) NOT NULL,  
    USER_NAME VARCHAR2(30),  
    GENDER VARCHAR2(10),  
    PHONE VARCHAR2(30),  
    EMAIL VARCHAR2(50),  
    GRADE_CODE NUMBER FOREIGN KEY REFERENCES USER_GRADE (GRADE_CODE)  
);
```

FOREIGN KEY

```
INSERT INTO USER_FOREIGNKEY  
VALUES(1, 'user01', 'pass01', '홍길동', '남', '010-1234-5678', 'hong123@kh.or.kr', 10);
```

1 행 미(가) 삽입되었습니다.

```
INSERT INTO USER_FOREIGNKEY  
VALUES(2, 'user02', 'pass02', '이순신', '남', '010-5678-9012', 'lee123@kh.or.kr', 20);
```

1 행 미(가) 삽입되었습니다.

```
INSERT INTO USER_FOREIGNKEY  
VALUES(3, 'user03', 'pass03', '유관순', '여', '010-9999-3131', 'yoo123@kh.or.kr', 30);
```

1 행 미(가) 삽입되었습니다.

```
INSERT INTO USER_FOREIGNKEY  
VALUES(4, 'user04', 'pass04', '안중근', '남', '010-2222-1111', 'ahn123@kh.or.kr', NULL);
```

1 행 미(가) 삽입되었습니다.

```
INSERT INTO USER_FOREIGNKEY  
VALUES(5, 'user05', 'pass05', '윤봉길', '남', '010-6666-1234', 'yoon123@kh.or.kr', 50);
```

오류 보고 -

SQL 오류: ORA-02291: integrity constraint (EMPLOYEE.SYS_C007202) violated - parent key not found

02291. 00000 - "integrity constraint (%s.%s) violated - parent key not found"

*Cause: A foreign key value has no matching primary key value.

*Action: Delete the foreign key or add a matching primary key.

FOREIGN KEY

<USER_GRADE TABLE>

GRADE_CODE	GRADE_NAME
10	일반회원
20	우수회원
30	특별회원

<USER_FOREIGNKEY TABLE>

USER_NO	USER_ID	USER_PWD	USER_NAME	GENDER	PHONE	EMAIL	GRADE_CODE
1	1 user01	pass01	홍길동	남	010-1234-5678	hong123@kh.or.kr	10
2	2 user02	pass02	이순신	남	010-5678-9012	lee123@kh.or.kr	20
3	3 user03	pass03	유관순	여	010-9999-3131	yoo123@kh.or.kr	30
4	4 user04	pass04	안중근	남	010-2222-1111	ahn123@kh.or.kr	(null)

- ** FOREIGN KEY 제약조건으로 USER_GRADE TABLE의 GRADE_CODE 컬럼을 참조한다.
- ** USER_GRADE 테이블을 USER_FOREIGNKEY 테이블에서 참조하는 관계이기 때문에, USER_GRADE 테이블의 데이터 삭제 시 참조무결성에 위배되기 때문에 삭제가 불가능하다.

삭제 옵션

부모 테이블의 데이터 삭제 시 자식 테이블의 데이터를 어떠한 방식으로 처리할지에 대한 내용을 제약조건 설정 시 옵션으로 지정할 수 있다.
기본 삭제 옵션은 ON DELETE RESTRICTED로 지정되어 있다.

FOREIGN KEY

```
CREATE TABLE USER_FOREIGNKEY(  
    USER_NO NUMBER PRIMARY KEY,  
    USER_ID VARCHAR2(20) UNIQUE,  
    USER_PWD VARCHAR2(30) NOT NULL,  
    USER_NAME VARCHAR2(30),  
    GENDER VARCHAR2(10),  
    PHONE VARCHAR2(30),  
    EMAIL VARCHAR2(50),  
    GRADE_CODE NUMBER REFERENCES USER_GRADE (GRADE_CODE)  
                                ON DELETE SET NULL  
);
```

```
DELETE FROM USER_GRADE  
WHERE GRADE_CODE = 10;
```

	GRADE_CODE	GRADE_NAME
1	20	우수회원
2	30	특별회원

	USER_NO	USER_ID	USER_PWD	USER_NAME	GENDER	PHONE	EMAIL	GRADE_CODE
1	1	user01	pass01	홍길동	남	010-1234-5678	hong123@kh.or.kr	(null)
2	2	user02	pass02	이순신	남	010-5678-9012	lee123@kh.or.kr	20
3	3	user03	pass03	유관순	여	010-9999-3131	yoo123@kh.or.kr	30
4	4	user04	pass04	안중근	남	010-2222-1111	ahn123@kh.or.kr	(null)

** 부모 테이블의 데이터 삭제 시 참조하고 있는 테이블의 컬럼 값이 NULL로 변경된다.

FOREIGN KEY

```
CREATE TABLE USER_FOREIGNKEY(  
    USER_NO NUMBER PRIMARY KEY,  
    USER_ID VARCHAR2(20) UNIQUE,  
    USER_PWD VARCHAR2(30) NOT NULL,  
    USER_NAME VARCHAR2(30),  
    GENDER VARCHAR2(10),  
    PHONE VARCHAR2(30),  
    EMAIL VARCHAR2(50),  
    GRADE_CODE NUMBER REFERENCES USER_GRADE (GRADE_CODE)  
                                ON DELETE CASCADE  
);
```

```
DELETE FROM USER_GRADE  
WHERE GRADE_CODE = 10;
```

	GRADE_CODE	GRADE_NAME
1	20	우수회원
2	30	특별회원

	USER_ID	USER_PWD	USER_NAME	GENDER	PHONE	EMAIL	GRADE_CODE
1	2 user02	pass02	이순신	남	010-5678-9012	lee123@kh.or.kr	20
2	3 user03	pass03	유관순	여	010-9999-3131	yoo123@kh.or.kr	30
3	4 user04	pass04	안중근	남	010-2222-1111	ahn123@kh.or.kr	(null)

** 부모 테이블의 데이터 삭제 시 참조하고 있는 테이블의 컬럼 값이 존재하던 행 전체를 삭제한다.

DDL(Data Definition Language) KH KH정보교육원

CHECK

해당 컬럼에 입력되거나 수정되는 값을 체크하여, 설정된 값 이외의 값이면 에러를 발생시킨다. 비교연산자를 이용하여 조건을 설정하며, 비교값은 리터럴만 사용 가능하고 변하는 값이나 함수 사용이 불가능하다.

```
CREATE TABLE USER_CHECK(  
    USER_NO NUMBER PRIMARY KEY,  
    USER_ID VARCHAR2(20) UNIQUE,  
    USER_PWD VARCHAR2(30) NOT NULL,  
    USER_NAME VARCHAR2(30),  
    GENDER VARCHAR2(10) CHECK (GENDER IN ('남', '여')),  
    PHONE VARCHAR2(30),  
    EMAIL VARCHAR2(50),  
    GRADE_CODE NUMBER  
);
```

```
INSERT INTO USER_CHECK  
VALUES(1, 'user01', 'pass01', '홍길동', '남자',  
      '010-1234-5678', 'hong123@kh.or.kr');
```

오류 보고 -

SQL 오류: ORA-02290: check constraint (EMPLOYEE.SYS_C007225) violated
02290. 00000 - "check constraint (%s.%s) violated"

*Cause: The values being inserted do not satisfy the named check

*Action: do not insert values that violate the constraint.

SUBQUERY를 이용한 CREATE TABLE

서브쿼리를 이용해서 SELECT의 조회 결과로 테이블을 생성하는 방법이다.
컬럼명과 데이터타입, 값이 복사되고, 제약조건은 NOT NULL만 복사된다.

```
CREATE TABLE EMPLOYEE_COPY  
AS SELECT EMP_ID, EMP_NAME, SALARY, DEPT_TITLE, JOB_NAME  
FROM EMPLOYEE  
LEFT JOIN DEPARTMENT ON(DEPT_CODE = DEPT_ID)  
LEFT JOIN JOB USING(JOB_CODE);
```

SQL | 인출된 모든 행: 23(0,016초)

EMP_ID	EMP_NAME	SALARY	DEPT_TITLE	JOB_NAME
1 214	방명수	1380000	인사관리부	사원
2 216	차태연	2780000	인사관리부	대리
3 217	전지연	3660000	인사관리부	대리
4 219	임시환	1550000	회계관리부	차장
5 220	이중석	2490000	회계관리부	차장
6 221	유하진	2480000	회계관리부	차장
7 206	백 이사	1800000	해외영업1부	사원
...				
21 200	선동일	8000000	총무부	대표
22 218	이오리	2890000	(null)	사원
23 213	하동운	2320000	(null)	대리