

Chap14. 제약 조건

- ▶ 14-1 제약 조건 종류
- ▶ 14-2 빈값을 허락하지 않는 NOT NULL
- ▶ 14-3 중복되지 않는 값 UNIQUE
- ▶ 14-4 유일하게 하나만 있는 값 PRIMARY KEY
- ▶ 14-5 다른 테이블과 관계를 맺는 FOREIGN KEY
- ▶ 14-6 데이터 형태와 범위를 정하는 CHECK
- ▶ 14-7 기본값을 정하는 DEFAULT

14-1 제약 조건 종류

▶ 제약 조건이란?

▶ 테이블 열에 저장될 데이터의 특성, 조건을 지정

종류	설명
NOT NULL	지정한 열에 NULL을 허용하지 않습니다. NULL을 제외한 데이터의 중복은 허용됩니다.
UNIQUE	지정한 열이 유일한 값을 가져야 합니다. 즉 중복될 수 없습니다. 단 NULL은 값의 중복에서 제외됩니다.
PRIMARY KEY	지정한 열이 유일한 값이면서 NULL을 허용하지 않습니다. PRIMARY KEY는 테이블에 하나만 지정 가능합니다.
FOREIGN KEY	다른 테이블의 열을 참조하여 존재하는 값만 입력할 수 있습니다.
CHECK	설정한 조건식을 만족하는 데이터만 입력 가능합니다.

14-1 제약 조건 종류

▶ 데이터 무결성이란?

데이터 무결성(data integrity)은 데이터베이스에 저장되는 데이터의 정확성과 일관성을 보장한다는 의미이며 이를 위해 항상 유지해야 하는 기본 규칙을 가지고 있습니다. 제약 조건은 이러한 데이터 무결성을 지키기 위한 안전장치로서 중요한 역할을 합니다. 그리고 테이블 데이터의 삽입(insert), 수정(update), 삭제(delete) 등 모든 과정에서 지켜야 합니다. 다음 표는 데이터 무결성의 종류를 정리해 놓은 것입니다.

종류	설명
영역 무결성 (domain integrity)	열에 저장되는 값의 적정 여부를 확인. 자료형, 적절한 형식의 데이터, NULL 여부같은 정해 놓은 범위를 만족하는 데이터임을 규정
개체 무결성 (entity integrity)	테이블 데이터를 유일하게 식별할 수 있는 기본키는 반드시 값을 가지고 있어야 하며 NULL이 될 수 없고 중복될 수도 없음을 규정
참조 무결성 (referential integrity)	참조 테이블의 외래키 값은 참조 테이블의 기본키로서 존재해야 하며 NULL이 가능

☺ 데이터 무결성의 자세한 내용은 관계형 데이터 모델 관련 서적 및 자료를 참고하세요.

이러한 무결성을 보장하기 위해 오라클에서는 앞에서 살펴본 다섯 가지 제약 조건을 제공합니다. 제약 조건은 데이터베이스 설계 시점, 즉 테이블을 생성할 때 주로 지정합니다. 하지만 테이블 생성 후에도 추가·변경·삭제할 수 있습니다. 따라서 제약 조건은 데이터 정의어(DDL)에서 활용합니다.

14-2 빈값을 허락하지 않는 NOT NULL

▶ NOT NULL

- ▶ 지정된 열에 **NULL 저장 불가**
- ▶ NULL외 **데이터의 중복은 허용**

실습 14-1 테이블을 생성할 때 NOT NULL 설정하기

```
01 CREATE TABLE TABLE_NOTNULL(  
02     LOGIN_ID VARCHAR2(20) NOT NULL,  
03     LOGIN_PWD VARCHAR2(20) NOT NULL,  
04     TEL VARCHAR2(20)  
05 );
```

```
06 DESC TABLE_NOTNULL;
```

:: 결과 화면(일부 열만 표시함)

Column Name	ID	PK	Index Pos	Null?	Data Type	Default	Histogram
LOGIN_ID	1			N	VARCHAR2 (20 Byte)		None
LOGIN_PWD	2			N	VARCHAR2 (20 Byte)		None
TEL	3			Y	VARCHAR2 (20 Byte)		None

실습 14-2 제약 조건이 NOT NULL인 열에 NULL 값 넣어보기

```
01 INSERT INTO TABLE_NOTNULL (LOGIN_ID, LOGIN_PWD, TEL)  
02 VALUES ('TEST_ID_01', NULL, '010-1234-5678');
```

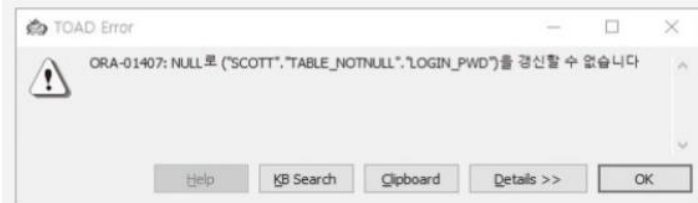
:: 결과 화면



실습 14-4 NOT NULL 제약 조건이 지정된 열 데이터를 NULL 값으로 업데이트하기

```
01 UPDATE TABLE_NOTNULL  
02     SET LOGIN_PWD = NULL  
03     WHERE LOGIN_ID = 'TEST_ID_01';
```

:: 결과 화면



14-2 빈값을 허락하지 않는 NOT NULL

▶ 제약 조건 확인

▶ USER_CONSTRAINTS

열 이름	설명
OWNER	제약 조건 소유 계정
CONSTRAINT_NAME	제약 조건 이름(직접 지정하지 않을 경우 오라클이 자동으로 지정함)
CONSTRAINT_TYPE	제약 조건 종류 C : CHECK, NOT NULL U : UNIQUE P : PRIMARY KEY R : FOREIGN KEY
TABLE_NAME	제약 조건을 지정한 테이블 이름

14-2 빈값을 허락하지 않는 NOT NULL

- ▶ 제약 조건 지정
 - ▶ 테이블을 생성하며 제약조건 지정

실습 14-6 테이블을 생성할 때 제약 조건에 이름 지정하기

```
01 CREATE TABLE TABLE_NOTNULL2(  
02     LOGIN_ID  VARCHAR2(20) CONSTRAINT TBLNN2_LGNID_NN NOT NULL,  
03     LOGIN_PWD VARCHAR2(20) CONSTRAINT TBLNN2_LGNPW_NN NOT NULL,  
04     TEL       VARCHAR2(20)  
05 );  
06  
07 SELECT OWNER, CONSTRAINT_NAME, CONSTRAINT_TYPE, TABLE_NAME  
08 FROM USER_CONSTRAINTS;
```

:: 결과 화면

OWNER	CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME
SCOTT	SYS_C0015544	C	TABLE_NOTNULL
SCOTT	SYS_C0015545	C	TABLE_NOTNULL
SCOTT	TBLNN2_LGNID_NN	C	TABLE_NOTNULL2
SCOTT	TBLNN2_LGNPW_NN	C	TABLE_NOTNULL2
SCOTT	FK_DEPTNO	R	EMP
SCOTT	PK_DEPT	P	DEPT
SCOTT	PK_EMP	P	EMP

14-2 빈값을 허락하지 않는 NOT NULL

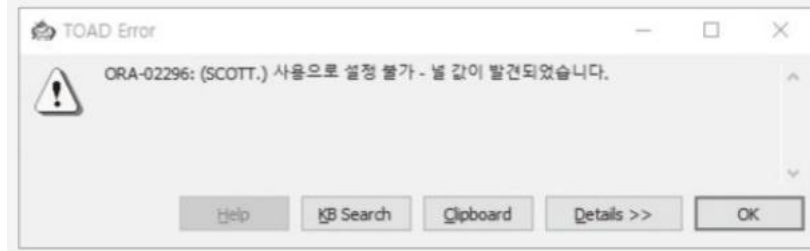
- ▶ 제약 조건 지정
 - ▶ 이미 생성한 테이블에 제약 조건 추가하기

LOGIN_ID	LOGIN_PWD	TEL
TEST_ID_01	1234	

실습 14-7 TEL 열에 NOT NULL 제약 조건 추가하기

```
01 ALTER TABLE TABLE_NOTNULL
02 MODIFY(TEL NOT NULL);
```

:: 결과 화면



실습 14-8 TEL 열 데이터 수정하기

```
01 UPDATE TABLE_NOTNULL
02     SET TEL = '010-1234-5678'
03     WHERE LOGIN_ID = 'TEST_ID_01';

04 SELECT * FROM TABLE_NOTNULL;
```

:: 결과 화면

LOGIN_ID	LOGIN_PWD	TEL
TEST_ID_01	1234	010-1234-5678

14-2 빈값을 허락하지 않는 NOT NULL

▶ 제약 조건 지정

▶ 이미 생성한 테이블에 제약 조건 추가하기

실습 14-9 NOT NULL 제약 조건 추가하기

```
01 ALTER TABLE TABLE_NOTNULL  
02 MODIFY(TEL NOT NULL);
```

```
SELECT OWNER, CONSTRAINT_NAME, CONSTRAINT_TYPE, TABLE_NAME  
FROM USER_CONSTRAINTS;
```

:: 결과 화면

OWNER	CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME
SCOTT	SYS_C0015544	C	TABLE_NOTNULL
SCOTT	SYS_C0015545	C	TABLE_NOTNULL
SCOTT	TBLNN2_LGNID_NN	C	TABLE_NOTNULL2
SCOTT	TBLNN2_LGNPW_NN	C	TABLE_NOTNULL2
SCOTT	SYS_C0015548	C	TABLE_NOTNULL

CONSTRAINT_TYPE이
C면 NOT NULL 또는
CHECK라는 뜻입니다.

14-2 빈값을 허락하지 않는 NOT NULL

- ▶ 제약 조건 지정
 - ▶ 제약 조건 이름 직접 지정

실습 14-10 제약 조건에 이름 지정해서 추가하기

```
01 ALTER TABLE TABLE_NOTNULL2
02 MODIFY(TEL CONSTRAINT TBLNN_TEL_NN NOT NULL);

03 SELECT OWNER, CONSTRAINT_NAME, CONSTRAINT_TYPE, TABLE_NAME
04 FROM USER_CONSTRAINTS;
```

:: 결과 화면

OWNER	CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME
SCOTT	SYS_C0015544	C	TABLE_NOTNULL
SCOTT	SYS_C0015545	C	TABLE_NOTNULL
SCOTT	TBLNN2_LGNID_NN	C	TABLE_NOTNULL2
SCOTT	TBLNN2_LGNPW_NN	C	TABLE_NOTNULL2
SCOTT	SYS_C0015548	C	TABLE_NOTNULL
SCOTT	TBLNN_TEL_NN	C	TABLE_NOTNULL2

실습 14-11 TABLE_NOTNULL2 테이블 열 구조 확인하기

```
01 DESC TABLE_NOTNULL2;
```

:: 결과 화면(일부 열만 표시함)

Column Name	ID	PK	Index Pos	Null?	Data Type
LOGIN_ID	1			N	VARCHAR2 (20 Byte)
LOGIN_PWD	2			N	VARCHAR2 (20 Byte)
TEL	3			N	VARCHAR2 (20 Byte)

14-2 빈값을 허락하지 않는 NOT NULL

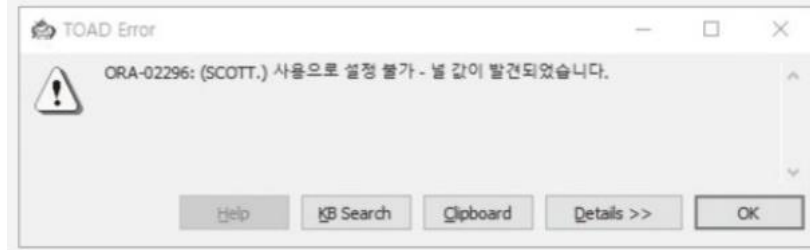
- ▶ 제약 조건 지정
 - ▶ 이미 생성한 테이블에 제약 조건 지정

LOGIN_ID	LOGIN_PWD	TEL
TEST_ID_01	1234	

실습 14-7 TEL 열에 NOT NULL 제약 조건 추가하기

```
01 ALTER TABLE TABLE_NOTNULL  
02 MODIFY(TEL NOT NULL);
```

:: 결과 화면



실습 14-8 TEL 열 데이터 수정하기

```
01 UPDATE TABLE_NOTNULL  
02 SET TEL = '010-1234-5678'  
03 WHERE LOGIN_ID = 'TEST_ID_01';  
  
04 SELECT * FROM TABLE_NOTNULL;
```

:: 결과 화면

LOGIN_ID	LOGIN_PWD	TEL
TEST_ID_01	1234	010-1234-5678

14-2 빈값을 허락하지 않는 NOT NULL

- ▶ 제약 조건 지정
 - ▶ 생성한 제약조건 이름 변경하기

실습 14-12 이미 생성된 제약 조건 이름 변경하기

```
01 ALTER TABLE TABLE_NOTNULL2
02 RENAME CONSTRAINT TBLNN_TEL_NN TO TBLNN2_TEL_NN;

03 SELECT OWNER, CONSTRAINT_NAME, CONSTRAINT_TYPE, TABLE_NAME
04 FROM USER_CONSTRAINTS;
```

:: 결과 화면

OWNER	CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME
SCOTT	SYS_C0015544	C	TABLE_NOTNULL
SCOTT	SYS_C0015545	C	TABLE_NOTNULL
SCOTT	TBLNN2_LGNID_NN	C	TABLE_NOTNULL2
SCOTT	TBLNN2_LGNPW_NN	C	TABLE_NOTNULL2
SCOTT	SYS_C0015548	C	TABLE_NOTNULL
SCOTT	TBLNN2_TEL_NN	C	TABLE_NOTNULL2
SCOTT	FK_DEPTNO	R	EMP
SCOTT	PK_DEPT	P	DEPT
SCOTT	PK_EMP	P	EMP

제약 조건 이름이
TBLNN_TEL_NN에서
TBLNN2_TEL_NN으로 바
뀌었습니다.

14-2 빈값을 허락하지 않는 NOT NULL

▶ 제약 조건 삭제

실습 14-13 제약 조건 삭제하기

```
01 ALTER TABLE TABLE_NOTNULL2
02 DROP CONSTRAINT TBLNN2_TEL_NN;

03 DESC TABLE_NOTNULL2;
```

제약 조건을 삭제하면 다음과 같이 TEL 열은 NULL이 저장될 수 있는 열이 됩니다.

:: 결과 화면(일부 열만 표시함)

Column Name	ID	PK	Index Pos	Null?	Data Type	Default	Histogram	Num Distinct
▶ LOGIN_ID	1			N	VARCHAR2 (20 Byte)		None	
LOGIN_PWD	2			N	VARCHAR2 (20 Byte)		None	
TEL	3			Y	VARCHAR2 (20 Byte)		None	

14-3 중복되지 않는 값 UNIQUE

▶ UNIQUE

- ▶ 지정된 열에 **중복 데이터 저장 불가**
- ▶ **NULL** 저장은 **가능**

실습 14-14 제약 조건 지정하기(테이블을 생성할 때)

```
01 CREATE TABLE TABLE_UNIQUE(  
02     LOGIN_ID  VARCHAR2(20) UNIQUE,  
03     LOGIN_PWD VARCHAR2(20) NOT NULL,  
04     TEL       VARCHAR2(20)  
05 );  
  
06 DESC TABLE_UNIQUE;
```

:: 결과 화면

Column Name	ID	PK	Index Pos	Null?
▶ LOGIN_ID	1		1	Y
LOGIN_PWD	2			N
TEL	3			Y

실습 14-21 테이블을 생성할 때 UNIQUE 제약 조건 설정하기

```
01 CREATE TABLE TABLE_UNIQUE2(  
02     LOGIN_ID  VARCHAR2(20) CONSTRAINT TBLUNQ2_LGNID_UNQ UNIQUE,  
03     LOGIN_PWD VARCHAR2(20) CONSTRAINT TBLUNQ2_LGNPW_NN NOT NULL,  
04     TEL       VARCHAR2(20)  
05 );
```

실습 14-22 생성한 UNIQUE 제약 조건 확인하기(USER_CONSTRAINTS 사용)

```
01 SELECT OWNER, CONSTRAINT_NAME, CONSTRAINT_TYPE, TABLE_NAME  
02     FROM USER_CONSTRAINTS  
03     WHERE TABLE_NAME LIKE 'TABLE_UNIQUE%';
```

:: 결과 화면(실습 14-22)

OWNER	CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME
▶ SCOTT	SYS_C0015610	C	TABLE_UNIQUE
SCOTT	SYS_C0015611	U	TABLE_UNIQUE
SCOTT	TBLUNQ2_LGNPW_NN	C	TABLE_UNIQUE2
SCOTT	TBLUNQ2_LGNID_UNQ	U	TABLE_UNIQUE2

14-3 중복되지 않는 값 UNIQUE

▶ UNIQUE

- ▶ 지정된 열에 **중복 데이터 저장 불가**
- ▶ **NULL** 저장은 **가능**

실습 14-16 TABLE_UNIQUE 테이블에 데이터 입력하기

```
01 INSERT INTO TABLE_UNIQUE(LOGIN_ID, LOGIN_PWD, TEL)
02 VALUES('TEST_ID_01', 'PWD01', '010-1234-5678');

03 SELECT * FROM TABLE_UNIQUE;
```

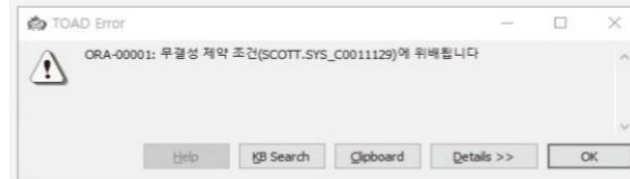
:: 결과 화면

LOGIN_ID	LOGIN_PWD	TEL
▶ TEST_ID_01	PWD01	010-1234-5678

실습 14-17 LOGIN_ID 열에 중복되는 데이터 넣기

```
01 INSERT INTO TABLE_UNIQUE (LOGIN_ID, LOGIN_PWD, TEL)
02 VALUES ('TEST_ID_01', 'PWD01', '010-1234-5678');
```

:: 결과 화면



14-3 중복되지 않는 값 UNIQUE

▶ UNIQUE

- ▶ 지정된 열에 **중복 데이터 저장 불가**
- ▶ **NULL** 저장은 **가능**

실습 14-18 TABLE_UNIQUE 테이블에 데이터 입력하기

```
01 INSERT INTO TABLE_UNIQUE(LOGIN_ID, LOGIN_PWD, TEL)
02 VALUES('TEST_ID_02', 'PWD01', '010-1234-5678');
```

```
SELECT * FROM TABLE_UNIQUE;
```

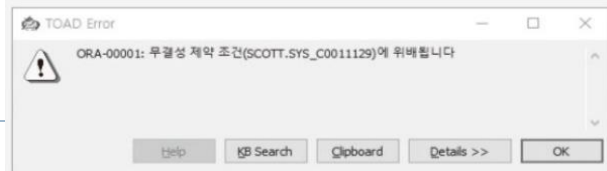
:: 결과 화면

LOGIN_ID	LOGIN_PWD	TEL
TEST_ID_01	PWD01	010-1234-5678
TEST_ID_02	PWD01	010-2345-6789

실습 14-20 TABLE_UNIQUE 테이블 데이터 수정하기

```
01 UPDATE TABLE_UNIQUE
02 SET LOGIN_ID='TEST_ID_01'
03 WHERE LOGIN_ID IS NULL;
```

:: 결과 화면



실습 14-19 UNIQUE 제약 조건이 지정된 열에 NULL 값 입력하기

```
01 INSERT INTO TABLE_UNIQUE(LOGIN_ID, LOGIN_PWD, TEL)
02 VALUES(NULL, 'PWD01', '010-2345-6789');
```

```
03 SELECT * FROM TABLE_UNIQUE;
```

:: 결과 화면

LOGIN_ID	LOGIN_PWD	TEL
TEST_ID_01	PWD01	010-1234-5678
TEST_ID_02	PWD01	010-2345-6789
	PWD01	010-2345-6789

14-3 중복되지 않는 값 UNIQUE

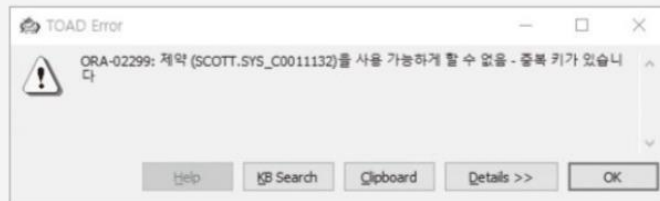
▶ UNIQUE

- ▶ 지정된 열에 **중복 데이터 저장 불가**
- ▶ **NULL** 저장은 **가능**

실습 14-23 이미 생성한 테이블 열에 UNIQUE 제약 조건 추가하기

```
01 ALTER TABLE TABLE_UNIQUE  
02 MODIFY(TEL UNIQUE);
```

:: 결과 화면



실습 14-24 TEL 열 값을 모두 NULL 값으로 변경하기

```
01 UPDATE TABLE_UNIQUE  
02     SET TEL = NULL;  
  
03 SELECT * FROM TABLE_UNIQUE;
```

:: 결과 화면

LOGIN_ID	LOGIN_PWD	TEL
TEST_ID_01	PWD01	
TEST_ID_02	PWD01	
	PWD01	

실습 14-25 TEL 값에 UNIQUE 제약 조건 설정하기

```
01 ALTER TABLE TABLE_UNIQUE  
02 MODIFY(TEL UNIQUE);
```

:: 결과 화면

OWNER	CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME
SCOTT	SYS_C0015610	C	TABLE_UNIQUE
SCOTT	SYS_C0015611	U	TABLE_UNIQUE
SCOTT	SYS_C0015616	U	TABLE_UNIQUE

CONSTRAINT_TYPE에
U가 설정되어 있습니다.
U는 UNIQUE 제약 조건을
의미합니다.

14-3 중복되지 않는 값 UNIQUE

▶ UNIQUE

실습 14-26 UNIQUE 제약 조건 이름 직접 지정하기

```
01 ALTER TABLE TABLE_UNIQUE2
02 MODIFY(TEL CONSTRAINT TBLUNQ_TEL_UNQ UNIQUE);

03 SELECT OWNER, CONSTRAINT_NAME, CONSTRAINT_TYPE, TABLE_NAME
04 FROM USER_CONSTRAINTS
05 WHERE TABLE_NAME LIKE 'TABLE_UNIQUE%';
```

:: 결과 화면

OWNER	CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME
SCOTT	SYS_C0015610	C	TABLE_UNIQUE
SCOTT	SYS_C0015611	U	TABLE_UNIQUE
SCOTT	SYS_C0015616	U	TABLE_UNIQUE
SCOTT	TBLUNQ2_LGNPW_NN	C	TABLE_UNIQUE2
SCOTT	TBLUNQ2_LGNID_UNQ	U	TABLE_UNIQUE2
SCOTT	TBLUNQ_TEL_UNQ	U	TABLE_UNIQUE2

실습 14-27 이미 만들어져 있는 UNIQUE 제약 조건 이름 수정하기

```
01 ALTER TABLE TABLE_UNIQUE2
02 RENAME CONSTRAINT TBLUNQ_TEL_UNQ TO TBLUNQ2_TEL_UNQ;

03 SELECT OWNER, CONSTRAINT_NAME, CONSTRAINT_TYPE, TABLE_NAME
04 FROM USER_CONSTRAINTS
05 WHERE TABLE_NAME LIKE 'TABLE_UNIQUE%';
```

:: 결과 화면

OWNER	CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME
SCOTT	SYS_C0015610	C	TABLE_UNIQUE
SCOTT	SYS_C0015611	U	TABLE_UNIQUE
SCOTT	SYS_C0015616	U	TABLE_UNIQUE
SCOTT	TBLUNQ2_LGNPW_NN	C	TABLE_UNIQUE2
SCOTT	TBLUNQ2_LGNID_UNQ	U	TABLE_UNIQUE2
SCOTT	TBLUNQ2_TEL_UNQ	U	TABLE_UNIQUE2

14-3 중복되지 않는 값 UNIQUE

▶ UNIQUE – 제약조건 삭제

실습 14-28 제약 조건 삭제하기

```
01  ALTER TABLE TABLE_UNIQUE2
02  DROP CONSTRAINT TBLUNQ2_TEL_UNQ;

03  SELECT OWNER, CONSTRAINT_NAME, CONSTRAINT_TYPE, TABLE_NAME
04  FROM USER_CONSTRAINTS
05  WHERE TABLE_NAME LIKE 'TABLE_UNIQUE%';
```

:: 결과 화면

	OWNER	CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME
▶	SCOTT	SYS_C0011128	C	TABLE_UNIQUE
	SCOTT	SYS_C0011129	U	TABLE_UNIQUE
	SCOTT	SYS_C0011133	U	TABLE_UNIQUE
	SCOTT	TBLUNQ2_LGNPW_NN	C	TABLE_UNIQUE2
	SCOTT	TBLUNQ2_LGNID_UNQ	U	TABLE_UNIQUE2

14-3 중복되지 않는 값 UNIQUE

▶ 제약 조건 확인 : USER_CONSTRAINTS

실습 14-15 USER_CONSTRAINTS 데이터 사전 뷰로 제약 조건 확인하기

```
01 SELECT OWNER, CONSTRAINT_NAME, CONSTRAINT_TYPE, TABLE_NAME
02    FROM USER_CONSTRAINTS
03    WHERE TABLE_NAME = 'TABLE_UNIQUE';
```

:: 결과 화면

OWNER	CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME
SCOTT	SYS_C0015610	C	TABLE_UNIQUE
SCOTT	SYS_C0015611	U	TABLE_UNIQUE

14-4 유일하게 하나만 있는 값 PRIMARY KEY

▶ PRIMARY KEY

- ▶ 지정된 열에 중복 데이터 저장 불가
- ▶ NULL 저장 불가
- ▶ 자동으로 인덱스 생성



14-4 유일하게 하나만 있는 값 PRIMARY KEY

▶ PRIMARY KEY

실습 14-29 테이블을 생성할 때 특정 열에 PRIMARY KEY 설정하기

```
01 CREATE TABLE TABLE_PK(  
02     LOGIN_ID VARCHAR2(20) PRIMARY KEY,  
03     LOGIN_PWD VARCHAR2(20) NOT NULL,  
04     TEL      VARCHAR2(20)  
05 );  
  
06 DESC TABLE_PK;
```

:: 결과 화면

Column Name	ID	PK	Index Pos	Null?
▶ LOGIN_ID	1	1	1	N
LOGIN_PWD	2			N
TEL	3			Y

실습 14-30 생성한 PRIMARY KEY 확인하기

```
01 SELECT OWNER, CONSTRAINT_NAME, CONSTRAINT_TYPE, TABLE_NAME  
02     FROM USER_CONSTRAINTS  
03     WHERE TABLE_NAME LIKE 'TABLE_PK%';
```

:: 결과 화면

OWNER	CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME
▶ SCOTT	SYS_C0015692	C	
SCOTT	SYS_C0015693	P	

CONSTRAINT_TYPE에 P가
설정되어 있습니다.
P는 PRIMARY KEY 제약 조
건을 의미합니다.

실습 14-31 생성한 PRIMARY KEY를 통해 자동 생성된 INDEX 확인하기

```
01 SELECT INDEX_NAME, TABLE_OWNER, TABLE_NAME  
02     FROM USER_INDEXES  
03     WHERE TABLE_NAME LIKE 'TABLE_PK%';
```

:: 결과 화면

INDEX_NAME	TABLE_OWNER	TABLE_NAME
▶ SYS_C0015693	SCOTT	TABLE_PK

14-4 유일하게 하나만 있는 값 PRIMARY KEY

▶ PRIMARY KEY

실습 14-32 제약 조건의 이름을 직접 지정하여 테이블 생성하기

```
01 CREATE TABLE TABLE_PK2(  
02     LOGIN_ID  VARCHAR2(20) CONSTRAINT TBLPK2_LGNID_PK PRIMARY KEY,  
03     LOGIN_PWD VARCHAR2(20) CONSTRAINT TBLPK2_LGNPW_NN NOT NULL,  
04     TEL       VARCHAR2(20)  
05 );  
  
06 DESC TABLE_PK2;
```

:: 결과 화면

Column Name	ID	PK	Index Pos	Null?	Data Type
▶ LOGIN_ID	1	1	1	N	VARCHAR2 (20 Byte)
LOGIN_PWD	2			N	VARCHAR2 (20 Byte)
TEL	3			Y	VARCHAR2 (20 Byte)

OWNER	CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME
▶ SCOTT	SYS_C0015692	C	TABLE_PK
SCOTT	SYS_C0015693	P	TABLE_PK
SCOTT	TBLPK2_LGNPW_NN	C	TABLE_PK2
SCOTT	TBLPK2_LGNID_PK	P	TABLE_PK2

INDEX_NAME	TABLE_OWNER	TABLE_NAME
▶ TBLPK2_LGNID_PK	SCOTT	TABLE_PK2
SYS_C0015693	SCOTT	TABLE_PK

14-4 유일하게 하나만 있는 값 PRIMARY KEY

▶ PRIMARY KEY – 중복 입력

실습 14-33 TABLE_PK 테이블에 데이터 입력하기

```
01 INSERT INTO TABLE_PK(LOGIN_ID, LOGIN_PWD, TEL)
02 VALUES('TEST_ID_01', 'PWD01', '010-1234-5678');

03 SELECT * FROM TABLE_PK;
```

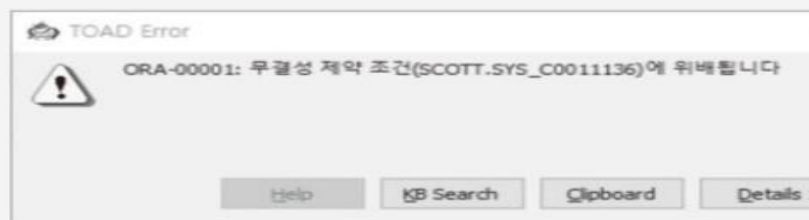
:: 결과 화면

LOGIN_ID	LOGIN_PWD	TEL
TEST_ID_01	PWD01	010-1234-5678

실습 14-34 TABLE_PK 테이블에 중복되는 데이터 입력하기

```
01 INSERT INTO TABLE_PK(LOGIN_ID, LOGIN_PWD, TEL)
02 VALUES('TEST_ID_01', 'PWD02', '010-2345-6789');
```

:: 결과 화면



PRIMARY KEY 제약 조건이 지정되어 있는 열은 중복을 허용하지 않습니다.

14-4 유일하게 하나만 있는 값 PRIMARY KEY

▶ PRIMARY KEY 생성방법 2

```
CREATE TABLE TABLE_NAME(  
    COL1 VARCHAR2(20) CONSTRAINT CONSTRAINT_NAME PRIMARY KEY,    -- 이름 지정함  
    COL2 VARCHAR2(20) NOT NULL,    -- 이름 지정하지 않음  
    COL3 VARCHAR2(20)  
);
```

```
CREATE TABLE TABLE_NAME(  
    COL1 VARCHAR2(20),  
    COL2 VARCHAR2(20),  
    COL3 VARCHAR2(20),  
    PRIMARY KEY (COL1),    -- 이름 지정하지 않음(COL1 열에 PRIMARY KEY 지정)  
    CONSTRAINT CONSTRAINT_NAME UNIQUE (COL2)    -- 이름 지정함(COL2 열에 UNIQUE 지정)  
);
```


14-5 다른 테이블과 관계를 맺는 FOREIGN KEY

▶ FOREIGN KEY

- ▶ 다른 테이블의 PRIMARY KEY를 참조
- ▶ 참조하고 있는 키의 데이터와 NULL만 저장 가능



14-5 다른 테이블과 관계를 맺는 FOREIGN KEY

▶ FOREIGN KEY

▶ 다른 테이블의 PRIMARY KEY를 참조

실습 14-37 EMP 테이블과 DEPT 테이블의 제약 조건 살펴보기

```
01 SELECT OWNER, CONSTRAINT_NAME, CONSTRAINT_TYPE, TABLE_NAME, R_OWNER, R_CONSTRAINT_NAME
02    FROM USER_CONSTRAINTS
03    WHERE TABLE_NAME IN ('EMP', 'DEPT');
```

:: 결과 화면

OWNER	CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME	R_OWNER	R_CONSTRAINT_NAME
SCOTT	PK_DEPT	P	DEPT		
SCOTT	PK_EMP	P	EMP		
SCOTT	FK_DEPTNO	(R)	EMP	SCOTT	PK_DEPT

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	1980-12-17	800		20
7499	ALLEN	SALESMAN	7698	1981-02-20	1600	300	30
7521	WARD	SALESMAN	7698	1981-02-22	1250	500	30
7566	JONES	MANAGER	7839	1981-04-02	2975		20
7654	MARTIN	SALESMAN	7698	1981-09-28	1250	1400	30
7698	BLAKE	MANAGER	7839	1981-05-01	2850		30
7782	CLARK	MANAGER	7839	1981-06-09	2450		10
7788	SCOTT	ANALYST	7566	1987-04-19	3000		20
7839	KING	PRESIDENT		1981-11-17	5000		10
7844	TURNER	SALESMAN	7698	1981-09-08	1500	0	30
7876	ADAMS	CLERK	7788	1987-05-23	1100		20
7900	JAMES	CLERK	7698	1981-12-03	950		30
7902	FORD	ANALYST	7566	1981-12-03	3000		20
7934	MILLER	CLERK	7782	1982-01-23	1300		10

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

14-5 다른 테이블과 관계를 맺는 FOREIGN KEY

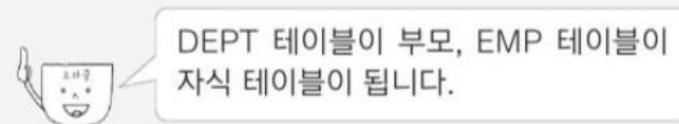
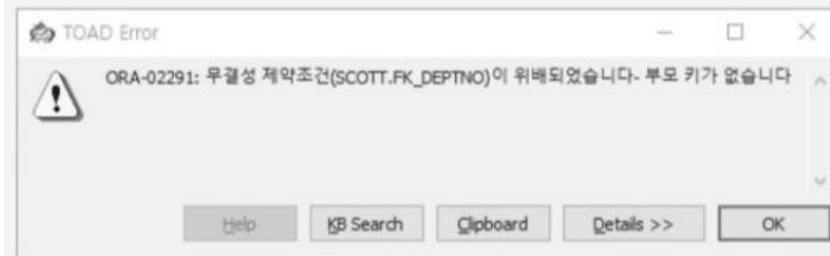
▶ FOREIGN KEY

- ▶ 참조하고 있는 키의 데이터와 NULL만 저장 가능

실습 14-38 FOREIGN KEY가 참조하는 열에 존재하지 않는 데이터 입력하기

```
01 INSERT INTO EMP(EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)
02 VALUES(9999, '홍길동', 'CLERK', '7788', TO_DATE('2017/04/30', 'YYYY/MM/DD'), 1200, NULL, 50);
```

:: 결과 화면



14-5 다른 테이블과 관계를 맺는 FOREIGN KEY

- ▶ 제약 조건 지정
 - ▶ 테이블을 생성하며 제약 조건 지정하기
 - ▶ 테이블을 생성하며 제약 조건 이름 직접 지정하기

실습 14-39 DEPT_FK 테이블 생성하기

```
01 CREATE TABLE DEPT_FK(  
02   DEPTNO NUMBER(2) CONSTRAINT DEPTFK_DEPTNO_PK PRIMARY KEY,  
03   DNAME  VARCHAR2(14),  
04   LOC    VARCHAR2(13)  
05 );
```

06 DESC DEPT_FK;

:: 결과 화면(일부 열만 표시함)

Column Name	ID	PK	Index Pos	Null?
▶ DEPTNO	1	1	1	N
DNAME	2			Y
LOC	3			Y

실습 14-40 EMP_FK 테이블 생성하기

```
01 CREATE TABLE EMP_FK(  
02   EMPNO      NUMBER(4) CONSTRAINT EMPFK_EMPNO_PK PRIMARY KEY,  
03   ENAME      VARCHAR2(10),  
04   JOB        VARCHAR2(9),  
05   MGR        NUMBER(4),  
06   HIREDATE   DATE,  
07   SAL        NUMBER(7,2),  
08   COMM       NUMBER(7,2),  
09   DEPTNO     NUMBER(2) CONSTRAINT EMPFK_DEPTNO_FK REFERENCES DEPT_FK (DEPTNO)  
10 );
```

11 DESC EMP_FK;

:: 결과 화면(일부 열만 표시함)

Column Name	ID	PK	Index Pos	Null?
▶ EMPNO	1	1	1	N
ENAME	2			Y
JOB	3			Y
MGR	4			Y
HIREDATE	5			Y
SAL	6			Y
COMM	7			Y
DEPTNO	8			Y

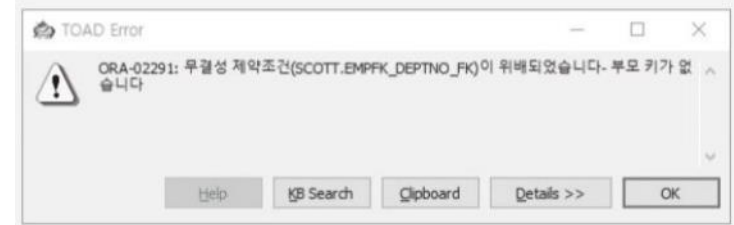
14-5 다른 테이블과 관계를 맺는 FOREIGN KEY

▶ 제약 조건 지정

실습 14-41 EMP_FK 테이블에 데이터 삽입하기(DEPTNO 데이터가 아직 DEPT_FK 테이블에 없을 때)

```
01 INSERT INTO EMP_FK
02 VALUES(9999, 'TEST_NNAME', 'TEST_JOB', NULL, TO_DATE('2001/01/01', 'YYYY/MM/DD'),
          3000, NULL, 10);
```

:: 결과 화면



실습 14-42 DEPT_FK에 데이터 삽입하기

```
01 INSERT INTO DEPT_FK
02 VALUES(10, 'TEST_DNAME', 'TEST_LOC');

03 SELECT * FROM DEPT_FK;
```

:: 결과 화면

DEPTNO	DNAME	LOC
10	TEST_DNAME	TEST_LOC

실습 14-43 EMP_FK 테이블에 데이터 삽입하기

```
01 INSERT INTO EMP_FK
02 VALUES(9999, 'TEST_NNAME', 'TEST_JOB', NULL, TO_DATE('2001/01/01', 'YYYY/MM/DD'),
          3000, NULL, 10);
```

:: 결과 화면

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
9999	TEST_NNAME	TEST_JOB		2001-01-01	3000		10

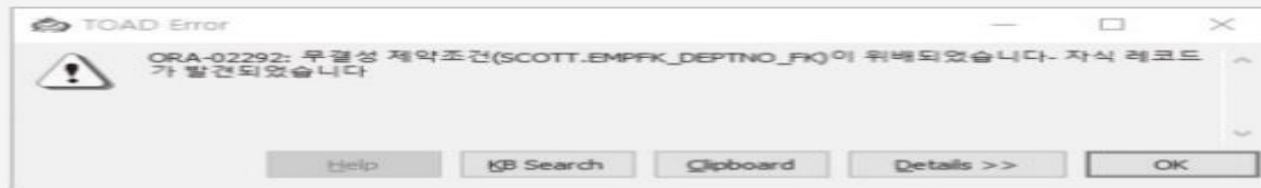
14-5 다른 테이블과 관계를 맺는 FOREIGN KEY

▶ 제약 조건 지정

실습 14-44 DEPT_FK 테이블의 10번 부서 데이터 삭제하기

```
01 DELETE FROM DEPT_FK
02 WHERE DEPTNO = 10;
```

:: 결과 화면



1. 현재 삭제하려는 열 값을 참조하는 데이터를 먼저 삭제한다.
ex) EMP_FK 테이블의 DEPTNO가 10번인 데이터를 삭제한 후 DEPT_FK 테이블의 10번 부서 삭제
2. 현재 삭제하려는 열 값을 참조하는 데이터를 수정한다.
ex) EMP_FK 테이블의 DEPTNO가 10번인 데이터를 다른 부서 번호 또는 NULL로 변경한 후 DEPT_FK 테이블의 10번 부서 삭제
3. 현재 삭제하려는 열을 참조하는 자식 테이블의 FOREIGN KEY 제약 조건을 해제한다.

14-5 다른 테이블과 관계를 맺는 FOREIGN KEY

- ▶ FOREIGN KEY로 참조 행 데이터 삭제하기
 - ▶ ON DELETE CASCADE

열 데이터를 삭제할 때 이 데이터를 참조하고 있는 데이터도 함께 삭제

CONSTRAINT [제약 조건 이름] REFERENCES 참조 테이블(참조할 열) ON DELETE CASCADE

DEPT_FK 테이블

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

10번 부서를 삭제할 경우

EMP_FK 테이블

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	1980-12-17	800		20
7499	ALLEN	SALESMAN	7698	1981-02-20	1600	300	30
7521	WARD	SALESMAN	7698	1981-02-22	1250	500	30
7566	JONES	MANAGER	7839	1981-04-02	2975		20
7654	MARTIN	SALESMAN	7698	1981-09-28	1250	1400	30
7698	BLAKE	MANAGER	7839	1981-05-01	2850		30
7782	CLARK	MANAGER	7839	1981-06-09	2450		10
7788	SCOTT	ANALYST	7566	1987-04-19	3000		20
7839	KING	PRESIDENT		1981-11-17	5000		10
7844	TURNER	SALESMAN	7698	1981-09-08	1500	0	30
7876	ADAMS	CLERK	7788	1987-05-23	1100		20
7900	JAMES	CLERK	7698	1981-12-03	950		30
7902	FORD	ANALYST	7566	1981-12-03	3000		20
7934	MILLER	CLERK	7782	1982-01-23	1300		10

10번 부서
에 속한 사
원 데이터
함께 삭제

14-5 다른 테이블과 관계를 맺는 FOREIGN KEY

- ▶ FOREIGN KEY로 참조 행 데이터 삭제하기
 - ▶ ON DELETE SET NULL

CONSTRAINT [제약 조건 이름] REFERENCES 참조 테이블(참조할 열) ON DELETE SET NULL

기본 형식

DEPT_FK 테이블

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

10번 부서를 삭제할 경우

EMP_FK 테이블

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	1980-12-17	800		20
7499	ALLEN	SALESMAN	7698	1981-02-20	1600	300	30
7521	WARD	SALESMAN	7698	1981-02-22	1250	500	30
7566	JONES	MANAGER	7839	1981-04-02	2975		20
7654	MARTIN	SALESMAN	7698	1981-09-28	1250	1400	30
7698	BLAKE	MANAGER	7839	1981-05-01	2850		30
7782	CLARK	MANAGER	7839	1981-06-09	2450		10
7788	SCOTT	ANALYST	7566	1987-04-19	3000		20
7839	KING	PRESIDENT		1981-11-17	5000		10
7844	TURNER	SALESMAN	7698	1981-09-08	1500	0	30
7876	ADAMS	CLERK	7788	1987-05-23	1100		20
7900	JAMES	CLERK	7698	1981-12-03	950		30
7902	FORD	ANALYST	7566	1981-12-03	3000		20
7934	MILLER	CLERK	7782	1982-01-23	1300		10

NULL

NULL

NULL

10번 부서에
속한 사원의
DEPTNO 열
값을 NULL
로 변경

14-6 데이터 형태와 범위를 정하는 CHECK

▶ CHECK

- ▶ 열에 저장할 수 있는 값의 범위 또는 패턴을 정의
- ▶ 조건식을 지정

실습 14-45 테이블을 생성할 때 CHECK 제약 조건 설정하기

```
01 CREATE TABLE TABLE_CHECK(  
02     LOGIN_ID VARCHAR2(20) CONSTRAINT TBLCK_LOGINID_PK PRIMARY KEY,  
03     LOGIN_PWD VARCHAR2(20) CONSTRAINT TBLCK_LOGINPW_CK CHECK (LENGTH(LOGIN_PWD) > 3),  
04     TEL      VARCHAR2(20)  
05 );  
  
06 DESC TABLE_CHECK;
```

 CHECK 제약 조건은 단순 연산뿐만 아니라 함수 활용도 가능합니다.

:: 결과 화면(일부 열만 표시함)

Column Name	ID	PK	Index Pos	Null?	Data Type
▶ LOGIN_ID	1	1	1	N	VARCHAR2 (20 Byte)
LOGIN_PWD	2			Y	VARCHAR2 (20 Byte)
TEL	3			Y	VARCHAR2 (20 Byte)

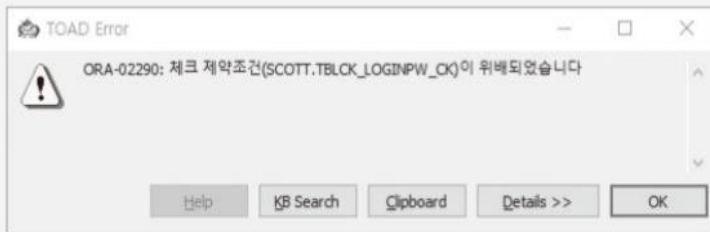
14-6 데이터 형태와 범위를 정하는 CHECK

▶ CHECK

실습 14-46 CHECK 제약 조건에 맞지 않는 예

```
01 INSERT INTO TABLE_CHECK
02 VALUES ('TEST_ID', '123', '010-1234-5678');
```

:: 결과 화면



실습 14-47 CHECK 제약 조건에 맞는 예

```
01 INSERT INTO TABLE_CHECK
02 VALUES ('TEST_ID', '1234', '010-1234-5678');
```

SELECT * FROM TABLE_CHECK;

:: 결과 화면

LOGIN_ID	LOGIN_PWD	TEL
TEST_ID	1234	010-1234-5678

실습 14-48 CHECK 제약 조건 확인하기

```
01 SELECT OWNER, CONSTRAINT_NAME, CONSTRAINT_TYPE, TABLE_NAME
02 FROM USER_CONSTRAINTS
03 WHERE TABLE_NAME LIKE 'TABLE_CHECK';
```

:: 결과 화면

OWNER	CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME
SCOTT	TBLCK_LOGINPW_CK	C	TABLE_CHECK
SCOTT	TBLCK_LOGINID_PK	P	TABLE_CHECK

14-7 기본값을 정하는 DEFAULT

▶ DEFAULT

- ▶ 저장값이 없을 경우 기본 값을 지정

실습 14-49 테이블을 생성할 때 DEFAULT 제약 조건 설정하기

```
01 CREATE TABLE TABLE_DEFAULT(  
02     LOGIN_ID  VARCHAR2(20) CONSTRAINT TBLCK2_LOGINID_PK PRIMARY KEY,  
03     LOGIN_PWD VARCHAR2(20) DEFAULT '1234',  
04     TEL       VARCHAR2(20)  
05 );
```

DESC TABLE_DEFAULT;

:: 결과 화면

Column Name	ID	PK	Index Pos	Null?	Data Type	Default	Histogram
▶ LOGIN_ID	1	1	1	N	VARCHAR2 (20 Byte)		None
LOGIN_PWD	2			Y	VARCHAR2 (20 Byte)	'1234'	None
TEL	3			Y	VARCHAR2 (20 Byte)		None

14-7 기본값을 정하는 DEFAULT

▶ DEFAULT

▶ 저장값이 없을 경우 기본 값을 지정


실습 14-50 DEFAULT로 지정한 기본값이 입력되는 INSERT문 확인하기

```
01 INSERT INTO TABLE_DEFAULT VALUES ('TEST_ID', NULL, '010-1234-5678');
02 INSERT INTO TABLE_DEFAULT (LOGIN_ID, TEL) VALUES ('TEST_ID2', '010-1234-5678');
```

```
SELECT * FROM TABLE_DEFAULT;
```

:: 결과 화면

	LOGIN_ID	LOGIN_PWD	TEL
▶	TEST_ID		010-1234-5678
	TEST_ID2	1234	010-1234-5678

 **한발 더 나가기!!** 제약 조건 비활성화, 활성화

ALTER TABLE 테이블 이름

DISABLE [NOVALIDATE / VALIDATE(선택)] **CONSTRAINT** 제약조건이름;

ALTER TABLE 테이블 이름

ENABLE [NOVALIDATE / VALIDATE(선택)] **CONSTRAINT** 제약조건이름;