

CONSTRAINT

강사 : 강병준

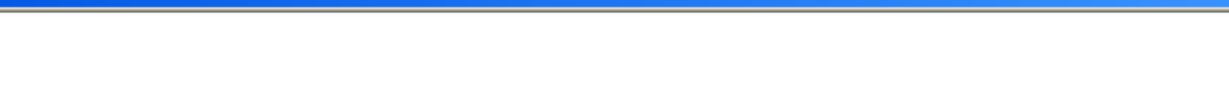
제약조건이란?

- 오라클 서버는 부적합한 데이터가 테이블에 삽입되는 것을 방지하기 위해 **CONSTRAINT** 를 사용
 - 테이블에서 행이 삽입, 갱신, 삭제될 때 마다 테이블에 설정된 규칙을 적용
 - 다른 테이블에 종속성이 있다면 테이블의 제거를 방지
- 데이터 무결성 제약 조건

제약조건	기 술
NOT NULL	이 열은 null 값을 포함하지 않음을 지정
UNIQUE KEY	테이블의 모든 행에 대해 유일해야 하는 값을 가진 열 또는 열의 조합을 지정
PRIMARY KEY	유일하게 테이블의 각 행을 식별
FOREIGN KEY	열과 참조된 테이블의 열 사이의 외래키 관계를 적용하고 설정
References	참조 무결성 제약조건 지정
CHECK	참이어야 하는 조건을 지정

제약 조건 확인하기

아래의 그림은 EMP 테이블에 INSERT 작업 중 무결성 제약 조건을 위반했을 때 나타나는 에러 메시지입니다.



The screenshot shows a Windows command prompt window with the title bar "C:\WINDOWS\system32\cmd.exe - sqlplus scott/tiger". The command prompt displays the following text:

```
SQL> INSERT INTO DEPT
  2  VALUES(10, 'TEST', 'SEOUL');
INSERT INTO DEPT
*
1행에 오류:
ORA-00001: 무결성 제약 조건 (SCOTT.PK_DEPT)에 위배됩니다
```

The error message "ORA-00001: 무결성 제약 조건 (SCOTT.PK_DEPT)에 위배됩니다" indicates a primary key constraint violation. This occurs because the department number 10 already exists in the DEPT table, and the SCOTT.PK_DEPT constraint requires that the DEPTNO column be unique.

DESC 명령어로는 NOT NULL 제약조건만 확인할 수 있고 DEPTNO 컬럼에 기본 키 제약 조건이 지정된 것을 알 수 없습니다.

```

C:\Windows\system32\cmd.exe - sqlplus scott/tiger

SQL> DESC DEPT
이름                널?       유형
-----
DEPTNO              NOT NULL   NUMBER(2)
DNAME                VCHAR2(14)
LOC                  VCHAR2(13)

SQL>

```

제약 조건 확인하기

- 오라클은 USER_CONSTRAINTS 데이터 디렉터리 뷰로 제약 조건에 관한 정보를 알려 줍니다.
- USER_CONSTRAINTS 데이터 디렉터리 뷰를 조회하면 내가 만든(USER) 제약 조건(CONSTRAINTS)의 정보를 조회 할 수 있습니다.

```
DESC USER_CONSTRAINTS;
```

제약 조건 확인하기

제약 조건을
소유한 사용자명

제약 조건 유형

CHECK 조건일
경우에는 어떤
내용이 조건으로
사용되었는지 설명

C:\WINDOWS\system32\cmd.exe - sqlplus scott/tiger

SQL> -- 데이터 디렉터리...

SQL> DESC USER_CONSTRAINTS

이름	널?	유형
OWNER	NOT NULL	VARCHAR2(30)
CONSTRAINT_NAME	NOT NULL	VARCHAR2(30)
CONSTRAINT_TYPE		VARCHAR2(1)
TABLE_NAME	NOT NULL	VARCHAR2(30)
SEARCH_CONDITION		LONG
R_OWNER		VARCHAR2(30)
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)

제약 조건 명

제약 조건이 속한 테이블 명

FOREIGN KEY 인 경우 어떤
PRIMARY KEY를 참조했는지에
대한 정보를 줌

제약 조건 확인하기

- USER_CONSTRAINTS 데이터 딕셔너리는 제약 조건의 정보를 위해서 많은 컬럼으로 구성되어 있습니다.
- OWNER는 제약 조건을 소유한 사용자명을 저장하는 컬럼입니다.
- CONSTRAINT_NAME은 제약 조건 명을
- CONSTRAINT_TYPE는 제약 조건 유형을 저장하는 컬럼입니다.
 - CONSTRAINT_TYPE은 P, R, U, C 4가지 값 중에 하나를 갖습니다.

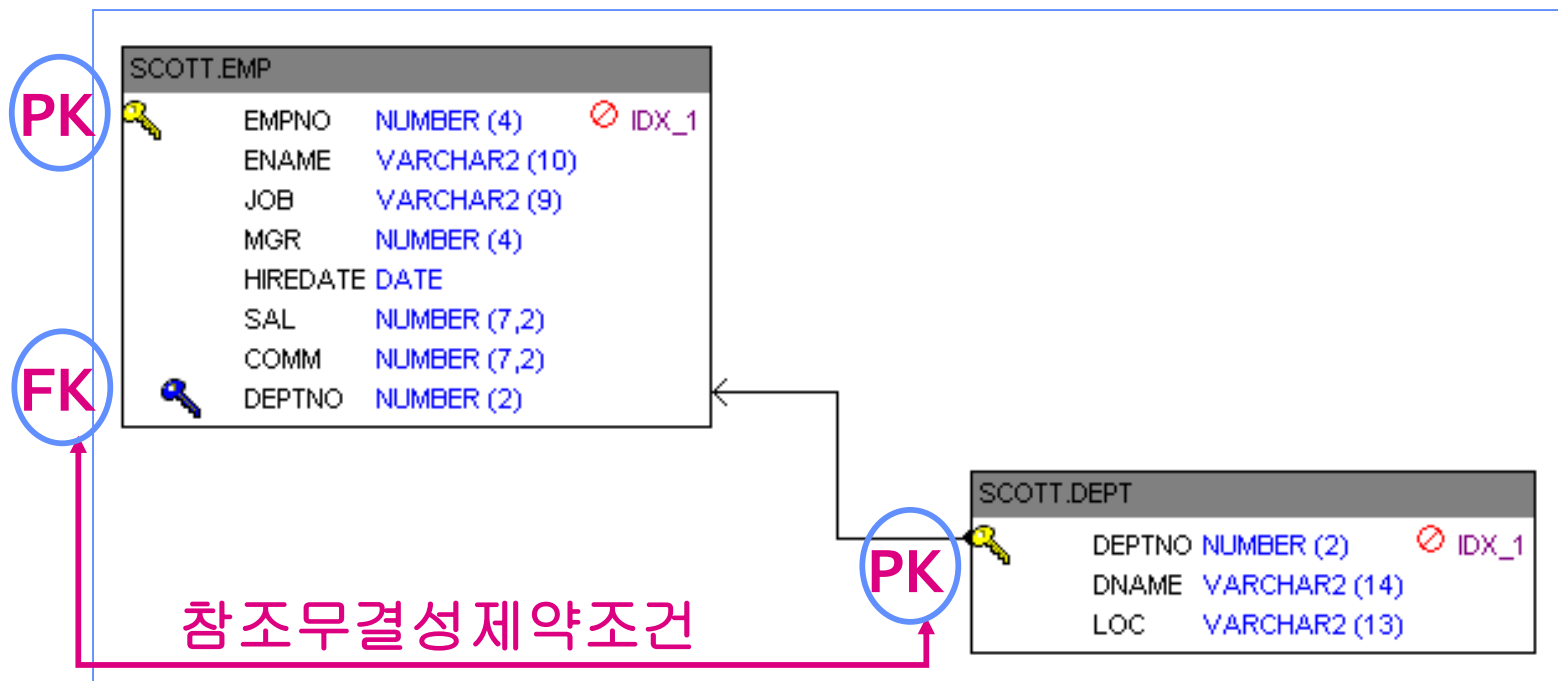
CONSTRAINT_TYPE	의미
P	PRIMARY KEY
R	FOREIGN KEY
U	UNIQUE
C	CHECK, NOT NULL

제약 조건 확인하기

- 제약 조건은 5개라고 했는데 제약 조건 유형은 4가지로 나타납니다.
- 제약 조건 중에 NOT NULL 은 컬럼에 NULL 값이 저장되어서는 안 된다는 조건입니다.
- NOT NULL 조건은 컬럼에 NULL 값을 체크하는 조건으로 처리되기 때문에 CHECK 를 나타내는 C로 표현됩니다.
- C는 CHECK 조건과 NOT NULL 조건을 모두 포함합니다.
- 제약 조건 유형은 제약 조건의 이니셜로 표현되지만 FOREIGN KEY 만은 R로 표현됩니다.
- FOREIGN KEY는 PRIMARY KEY를 참조하기 때문에 참조 무결성을 지켜야 합니다.
- 참조 무결성(REFERENCE INTEGRITY)의 이니셜인 R을 FOREIGN KEY 의 제약 조건 유형으로 사용합니다.
- TABLE_NAME은 각 제약 조건들이 속한 테이블의 이름입니다. 제약 조건 유형이 C인 경우에는 NOT NULL 조건과 CHECK 조건이 모두 포함되는데 NOT NULL 조건인 경우에는 따로 언급되는 내용이 없지만 CHECK 조건일 경우에는 어떤 내용이 조건으로 사용되었는지를 기록하고 있어야 하는데, SEARCH_CONDITION은 제약 조건 유형이 C인 경우 각 행에 대한 조건을 설명해 줍니다.
- R_CONSTRAINT_NAME은 제약 조건이 FOREIGN KEY인 경우 어떤 PRIMARY KEY를 참조 했는지에 대한 정보를 갖습니다.

제약조건 예

■ emp & dept의 Constraint



제약조건 지침

- 사용자가 제약조건을 명명하거나 오라클 서버는 SYS_Cn포맷을 사용하여 이름을 생성.
(n은 이름을 구별하기 위한 고유번호 임)
- 제약조건 생성
 - 테이블이 생성되는 시간과 동일
 - 테이블이 생성된 후에 정의될 수 있음
- 열 또는 테이블 레벨에서 제약조건을 정의

제약조건 지칭

```
SQL> CREATE TABLE SAWON (  
  2  S_NO NUMBER(4),  
  3  S_NAME VARCHAR2(10) NOT NULL,  
  4  S_HIREDATE DATE CONSTRAINT SAWON_S_HIREDATE_NN NOT NULL);
```

테이블이 생성되었습니다.

```
SQL> INSERT INTO SAWON  
  2  VALUES(1, '길동', NULL);  
INSERT INTO SAWON
```

*

1행에 오류:

ORA-01400: NULL을 ("SCOTT"."SAWON"."S_HIREDATE") 안에 삽입할 수 없습니다

위에서 SAWON 테이블의 S_NAME 컬럼과 S_HIREDATE 컬럼에 NOT NULL 제약조건이 지정되었으며, 해당 컬럼에 NULL 값을 입력하려고 시도하면 제약조건에 위배되어 오류가 발생한다. S_NAME 컬럼과 같이 제약조건 선언시 제약조건 이름을 지정하지 않으면 Oracle 서버가 임의로 제약조건 이름을 정의하게 된다. 제약조건을 검색하는 방법은 다음과 같다.

```
SQL> SELECT CONSTRAINT_NAME, CONSTRAINT_TYPE  
  2  FROM USER_CONSTRAINTS  
  3  WHERE TABLE_NAME = 'SAWON';
```

CONSTRAINT_NAME	C
SAWON_S_HIREDATE_NN	C
SYS_C003005	C

제약조건 정의

```
CREATE TABLE      [schema.]table  
                   (column datatype [DEFAULT expr]  
                   [column_constraint],  
                   .....  
                   [table_constraint ]);
```

- **Schema** 소유자 명과 동일
- **Table** 테이블의 이름
- **DEFAULT expr** 값이 INSERT 문장에서 생략되면 디폴트 값을 지정
- **Column** 열의 이름
- **Datatype** 열의 데이터 유형과 길이
- **Column_constraint** 열 정의의 일부분서 무결성 제약조건
- **Table_constraint** 테이블 정의의 일부분서 무결성 제약 조건

제약조건 정의

- 열 제약조건 레벨

- 열별로 정의 무결성 제약조건의 어떤 유형도 정의 가능

컬럼 데이터타입 [CONSTRAINT *constraint_name*] *constraint_type*

- 테이블 제약조건 레벨

- 하나 이상의 열을 참조하고, 테이블의 열 정의와는 개별적으로 정의, NOT NULL을 제외한 임의의 제약조건 정의 가능

Column, ...

[CONSTRAINT *constraint_name*] *constraint_type* (*column*, ...),

무결성 제약 조건 확인

■ USER_CONSTRAINTS

- 소유자, 무결성 제약 조건명, 무결성 제약 조건 종류 등을 확인

■ USER_CONS_COLUMNS

- 소유자, 무결성 제약 조건명, 테이블명, 컬럼명 등을 확인

제약조건 보기

무결성 제약 조건 표시
C : CHECK 또는 NOT NULL
P : PRIMARY KEY
U : UNIQUE KEY
F : FOREIGN KEY
R : 테이블간의 관계 설정
V : 뷰의 무결성제약조건 명시

- 모든 제약조건의 정의와 이름을 보기위해
USER_CONSTRAINTS 테이블을 질의함

```
SQL> SELECT      constraint_name, constraint_type,  
2                search_condition, status  
3      FROM        user_constraints  
4      WHERE       table_name = 'EMP';
```

CONSTRAINT_NAME	C	SEARCH_CONDITION	STATUS
SYS_C00674	C	EMPNO IS NOT NULL	ENABLE
SYS_C00675	C	DEPTNO IS NOT NULL	ENABLE
EMP_EMPNO_PK	P		ENABLE
...			

제약조건과 연관된 열 보기

- **USER_CONS_COLUMNS** 뷰에서 제약조건 이름과 관련된 열을 봄

```
SQL>  SELECT      constraint_name, column_name
      2  FROM        user_cons_columns
      3  WHERE       table_name = 'EMP';
```

CONSTRAINT_NAME	COLUMN_NAME
-----	-----
EMP_DEPTNO_FK	DEPTNO
EMP_EMPNO_PK	EMPNO
EMP_MGR_FK	MGR
SYS_C00674	EMPNO
SYS_C00675	DEPTNO

NOT NULL 제약조건

- 열에 대해 null 값이 허용되지 않도록 보증
- NOT NULL 제약조건이 없는 열은 디폴트로 null 값을 포함할 수 있음
- 테이블 레벨이 아닌 열 레벨에서 정의

```
SQL> CREATE TABLE emp_nn (  
2      empno  NUMBER(4),  
3      ename   VARCHAR2(10) NOT NULL,  
4      job     VARCHAR2(9),  
5      mgr     NUMBER(4),  
6      hiredate DATE,  
7      sal     NUMBER(7,2),  
8      comm    NUMBER(7,2),  
9      deptno  NUMBER(7,2) constraint emp_nn_deptno NOT NULL) ;
```

*Insert into emp_nn (empno, ename)
Values(1000, 'superman');
에러 발생!!!*

NOT NULL 제약조건

NOT NULL

NOT NULL 제약조건은 해당 컬럼에 NULL 값이 입력되지 않도록 제한하는 것으로, NOT NULL 제약조건이 정의되지 않은 컬럼은 디폴트로 NULL 값의 저장이 허용된다. NOT NULL 제약조건은 컬럼 수준에서만 지정가능하며 사용방법은 다음과 같다.

```
SQL> CREATE TABLE SAWON (  
  2  S_NO NUMBER(4),  
  3  S_NAME VARCHAR2(10) NOT NULL,  
  4  S_HIREDATE DATE CONSTRAINT SAWON_S_HIREDATE_NN NOT NULL);
```

테이블이 생성되었습니다.

```
SQL> INSERT INTO SAWON  
  2  VALUES(1, '길동', NULL);  
INSERT INTO SAWON
```

*

1행에 오류:

ORA-01400: NULL을 ("SCOTT"."SAWON"."S_HIREDATE") 안에 삽입할 수 없습니다

실습하기

사원 테이블(EMP02)을 사원번호, 사원명, 직급, 부서번호 4개의 칼럼으로 구성하되 이번에는 사원번호와 사원명에 NOT NULL 조건을 지정하도록 합니다. 제약 조건은 칼럼명과 자료형을 기술한 후에 연이어서 NOT NULL을 기술하면 됩니다.

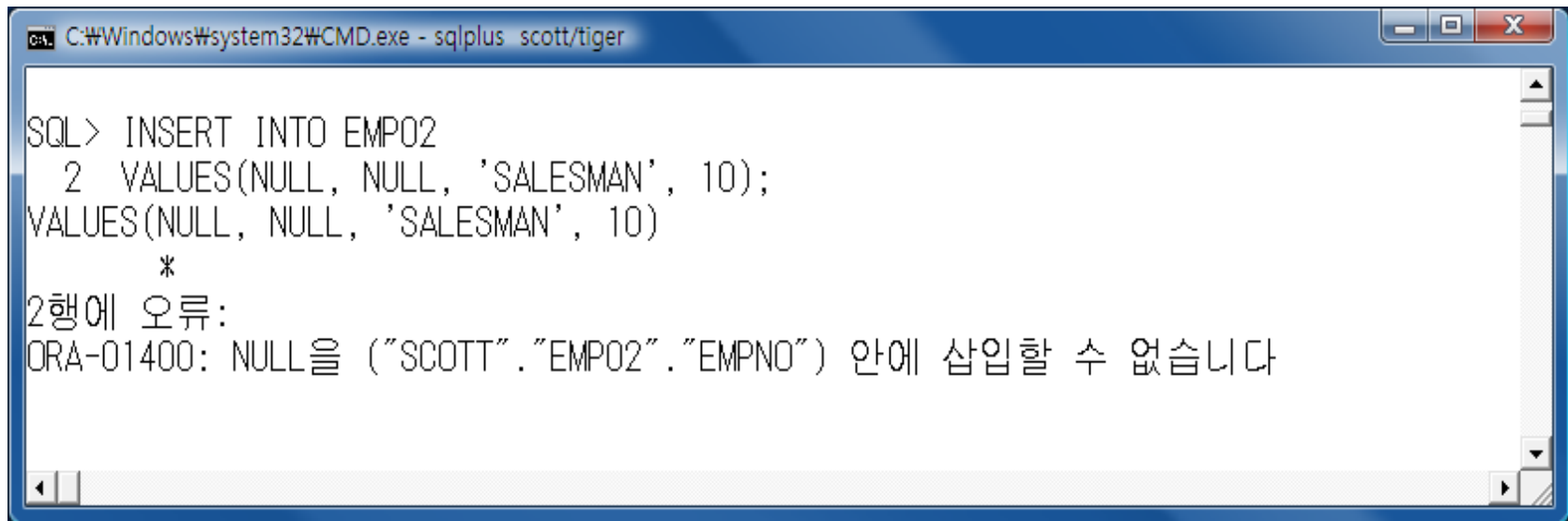
1. 지금까지 실습에 사용했던 사원 테이블과 유사한 구조의 사원번호, 사원명, 직급, 부서번호 4개의 칼럼으로 구성된 EMP02 테이블을 생성하되 EMPNO와 EMPNAME 컬럼에 NOT NULL 제약 조건 설정해 봅니다.

```
CREATE TABLE EMP02(  
  EMPNO NUMBER(4) NOT NULL,  
  ENAME VARCHAR2(10) NOT NULL,  
  JOB VARCHAR2(9),  
  DEPTNO NUMBER(2)  
);
```

실습하기

2. 위에서 생성한 EMP02 테이블에는 데이터를 추가해 봅시다.

```
INSERT INTO EMP02  
VALUES(NULL, NULL, 'SALESMAN', 10);
```

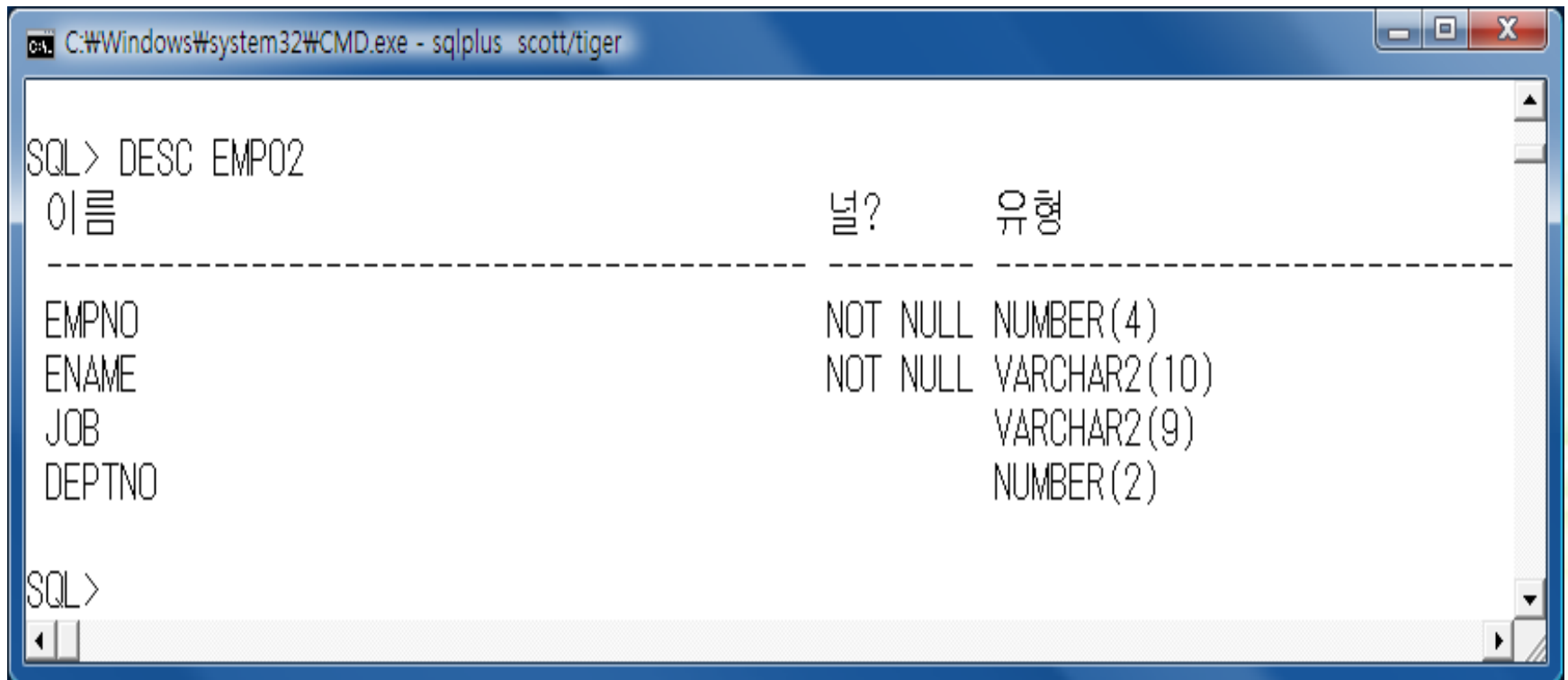
A screenshot of a Windows command prompt window. The title bar reads "C:\Windows\system32\CMD.exe - sqlplus scott/tiger". The command prompt shows the following text:
SQL> INSERT INTO EMP02
2 VALUES(NULL, NULL, 'SALESMAN', 10);
VALUES(NULL, NULL, 'SALESMAN', 10)
*
2행에 오류:
ORA-01400: NULL을 ("SCOTT"."EMP02"."EMPNO") 안에 삽입할 수 없습니다
The window has standard Windows window controls (minimize, maximize, close) in the top right corner and a scrollbar on the right side.

EMP02 테이블은 사원번호와 사원명에 NOT NULL 조건을 지정하였기에 사원번호에 NULL을 추가하는 명령어에서 오류가 발생합니다.

실습하기

3. DESC 명령어로 NOT NULL 제약조건이 설정되어 있음을 확인할 수 있습니다.

DESC EMP02



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\CMD.exe - sqlplus scott/tiger". The command "SQL> DESC EMP02" has been entered. The output displays the structure of the EMP02 table, including column names, nullability, and data types.

이름	널?	유형
EMPNO	NOT NULL	NUMBER(4)
ENAME	NOT NULL	VARCHAR2(10)
JOB		VARCHAR2(9)
DEPTNO		NUMBER(2)

UNIQUE KEY 제약조건

- UNIQUE key 무결성 제약조건은 열 또는 열의 집합의 모든 테이블 값들이 유일해야 함을 요구 즉 지정된 열 또는 열의 집합에서 중복 값을 가지는 테이블의 두 행은 없다
- UNIQUE key 제약조건에 정의에 포함된 열은 unique key라 부름
- UNIQUE key가 하나이상의 열을 포함한다면, 열 그룹은 composite unique key라 부름
- UNIQUE key 제약조건은 동일 열에 대해 NOT NULL 제약조건을 정의하지 않으면 null의 출력을 허용
 - 행은 null이 어느 것과도 동일하게 여겨지지 않기 때문에 NOT NULL 제약조건이 없으면 열에 대해 null을 포함할 수 있다
 - 열에서 null은 항상 UNIQUE key 제약조건을 만족함

UNIQUE key 제약조건

- 열 또는 테이블 레벨에서 정의될 수 있음
- 조합 유일 키는 테이블 레벨 정의를 사용하여 생성됨

```
SQL> CREATE TABLE deptuk (  
2         deptno  NUMBER(2),  
3         dname   VARCHAR2(14),  
4         loc     VARCHAR2(13),  
5         CONSTRAINT deptuk_dname_uk UNIQUE(dname));
```

UNIQUE key 제약조건

```
SQL> CREATE TABLE SAWON (  
2   S_NO NUMBER(4),  
3   S_NAME VARCHAR2(10),  
4   S_SAL NUMBER(10),  
5   S_EMAIL VARCHAR2(20) CONSTRAINT SAWON_S_EMAIL_UK UNIQUE);
```

테이블이 생성되었습니다.

```
SQL> INSERT INTO SAWON VALUES (1, '길동', 1000, 'GDH@XYZ.COM');
```

1 개의 행이 만들어졌습니다.

```
SQL> INSERT INTO SAWON VALUES (2, '콩쥐', 2000, 'GDH@XYZ.COM');  
INSERT INTO SAWON VALUES (2, '콩쥐', 2000, 'GDH@XYZ.COM')
```

*

1행에 오류:

ORA-00001: 무결성 제약 조건(SCOTT.SAWON_S_EMAIL_UK)에 위배됩니다

```
SQL> INSERT INTO SAWON VALUES (3, '팥쥐', 1500, NULL);
```

1 개의 행이 만들어졌습니다.

실습하기

다음은 사원 테이블의 사원번호를 유일키로 지정한 예입니다.

1. 지금까지 실습에 사용했던 사원 테이블과 유사한 구조의 사원번호, 사원명, 직급, 부서번호 4개의 칼럼으로 구성된 EMP03 테이블을 생성하되 사원번호를 유일키로 지정합니다. 제약 조건은 칼럼명과 자료형을 기술한 후에 연이어서 UNIQUE를 기술하면 됩니다.

```
CREATE TABLE EMP03(  
  EMPNO NUMBER(4) UNIQUE,  
  ENAME VARCHAR2(10) NOT NULL,  
  JOB VARCHAR2(9),  
  DEPTNO NUMBER(2)  
);
```

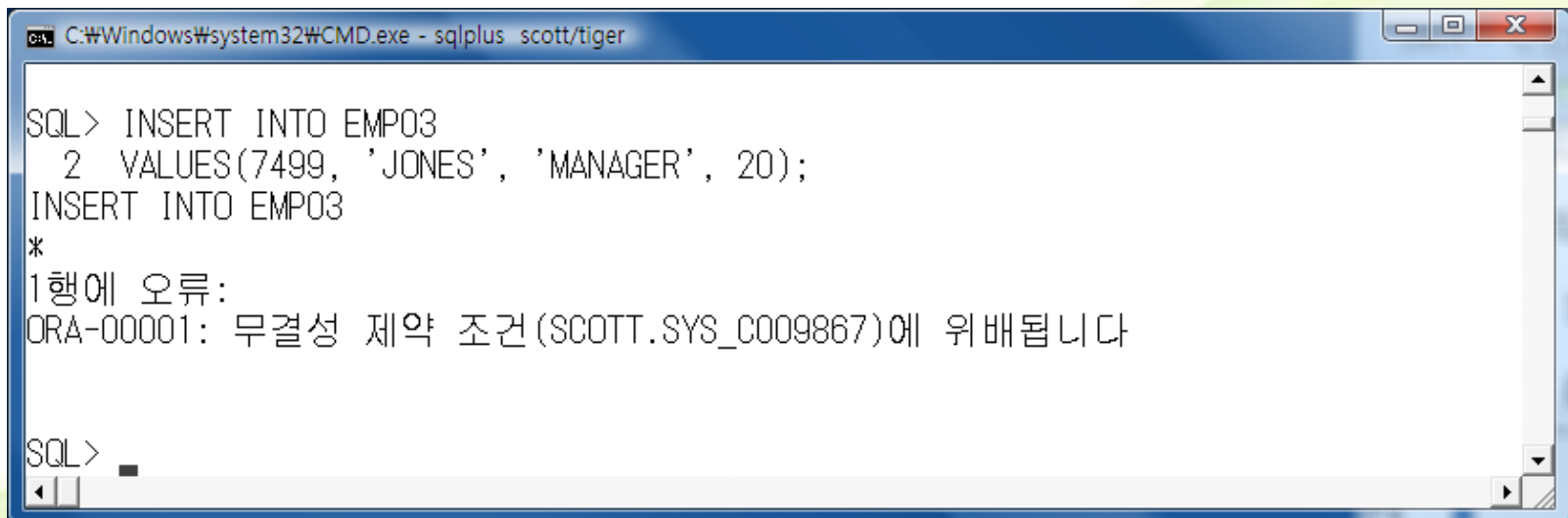

실습하기

2. 위에서 생성한 EMP03 테이블에 데이터를 추가해 봅시다.

```
INSERT INTO EMP03  
VALUES(7499, 'ALLEN', 'SALESMAN', 30);
```

3. 앞에서 사원번호로 7499번의 자료를 입력하였는데 다시 동일한 사원번호를 입력하면 어떻게 될까요?

```
INSERT INTO EMP03  
VALUES(7499, 'JONES', 'MANAGER', 20);
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\CMD.exe - sqlplus scott/tiger". The command prompt displays the following text:

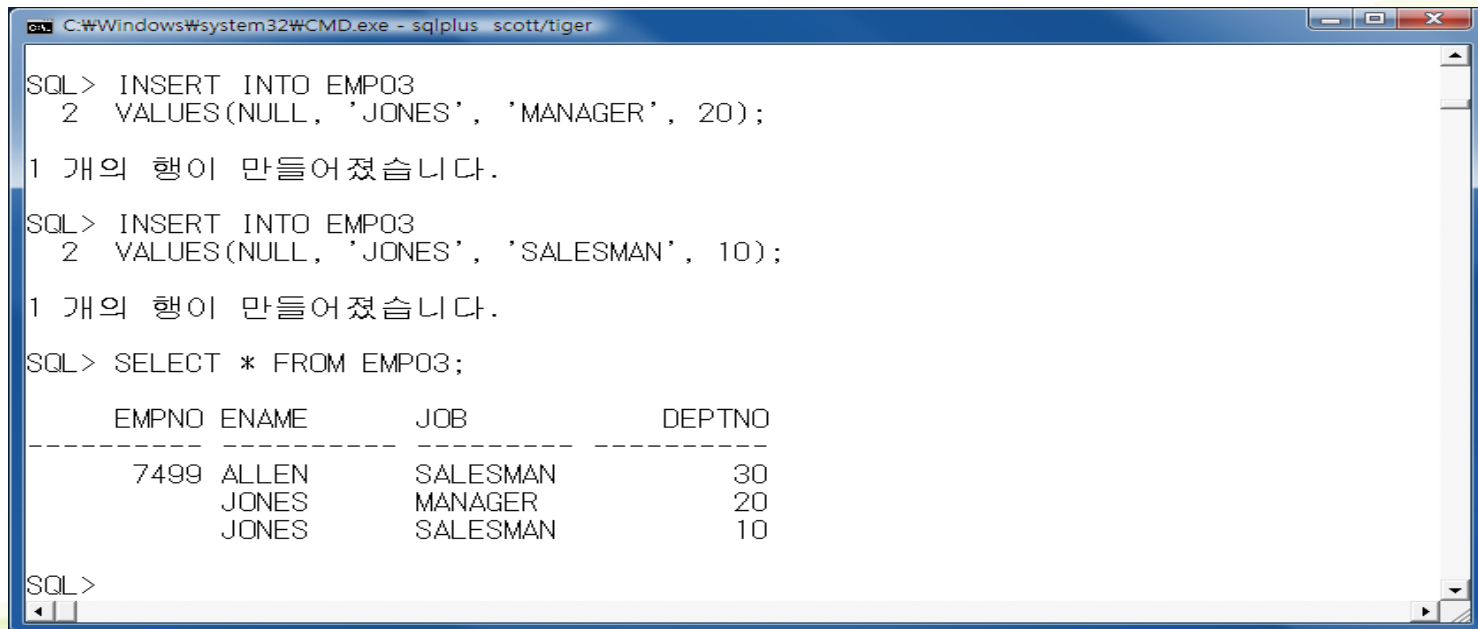
```
SQL> INSERT INTO EMP03  
2  VALUES(7499, 'JONES', 'MANAGER', 20);  
INSERT INTO EMP03  
*  
1행에 오류:  
ORA-00001: 무결성 제약 조건 (SCOTT.SYS_C009867)에 위배됩니다  
  
SQL>
```

실습하기

4. 하지만 NULL 값은 중복되어 저장할 수 있습니다. UNIQUE는 값(VALUE)이 유일함을 의미하는 것입니다. NULL은 값(VALUE)에서 제외되므로 유일한 조건인지를 체크하는 값에서 제외됩니다.

```
INSERT INTO EMP03  
VALUES(NULL, 'JONES', 'MANAGER', 20);
```

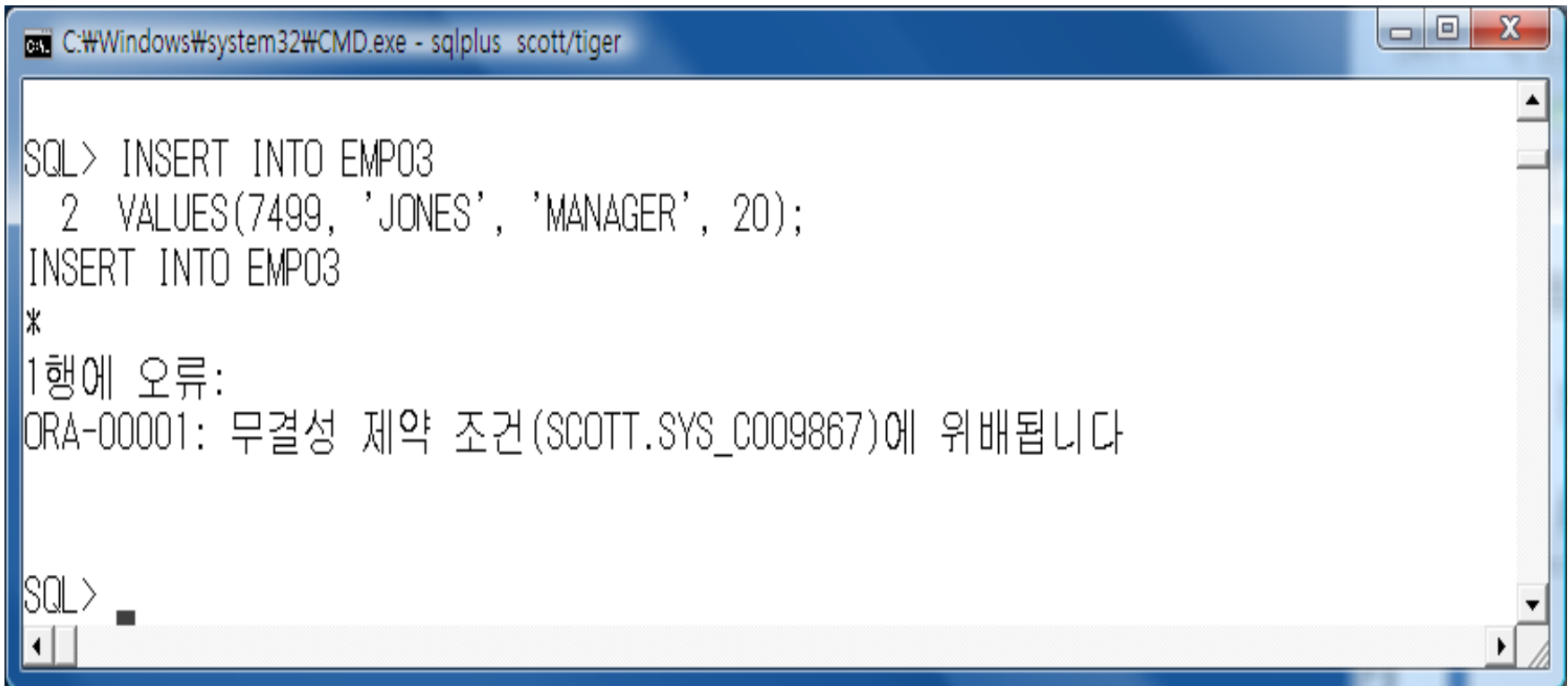
```
INSERT INTO EMP03  
VALUES(NULL, 'JONES', 'SALESMAN', 10);
```



```
C:\Windows\system32\CMD.exe - sqlplus scott/tiger  
  
SQL> INSERT INTO EMP03  
2  VALUES(NULL, 'JONES', 'MANAGER', 20);  
  
1 개의 행이 만들어졌습니다.  
  
SQL> INSERT INTO EMP03  
2  VALUES(NULL, 'JONES', 'SALESMAN', 10);  
  
1 개의 행이 만들어졌습니다.  
  
SQL> SELECT * FROM EMP03;  
  
      EMPNO ENAME      JOB              DEPTNO  
-----  
      7499 ALLEN      SALESMAN         30  
          JONES      MANAGER          20  
          JONES      SALESMAN         10  
  
SQL>
```

제약 조건명을 명시해서 제약 조건 설정하기

- 지금까지는 사용자가 제약 조건명을 지정하지 않고 제약 조건만을 명시했습니다.
- 이럴 경우 오라클 서버가 자동으로 제약 조건명을 부여합니다.
- 오라클이 부여하는 제약 조건명은 SYS_ 다음에 숫자를 나열합니다.



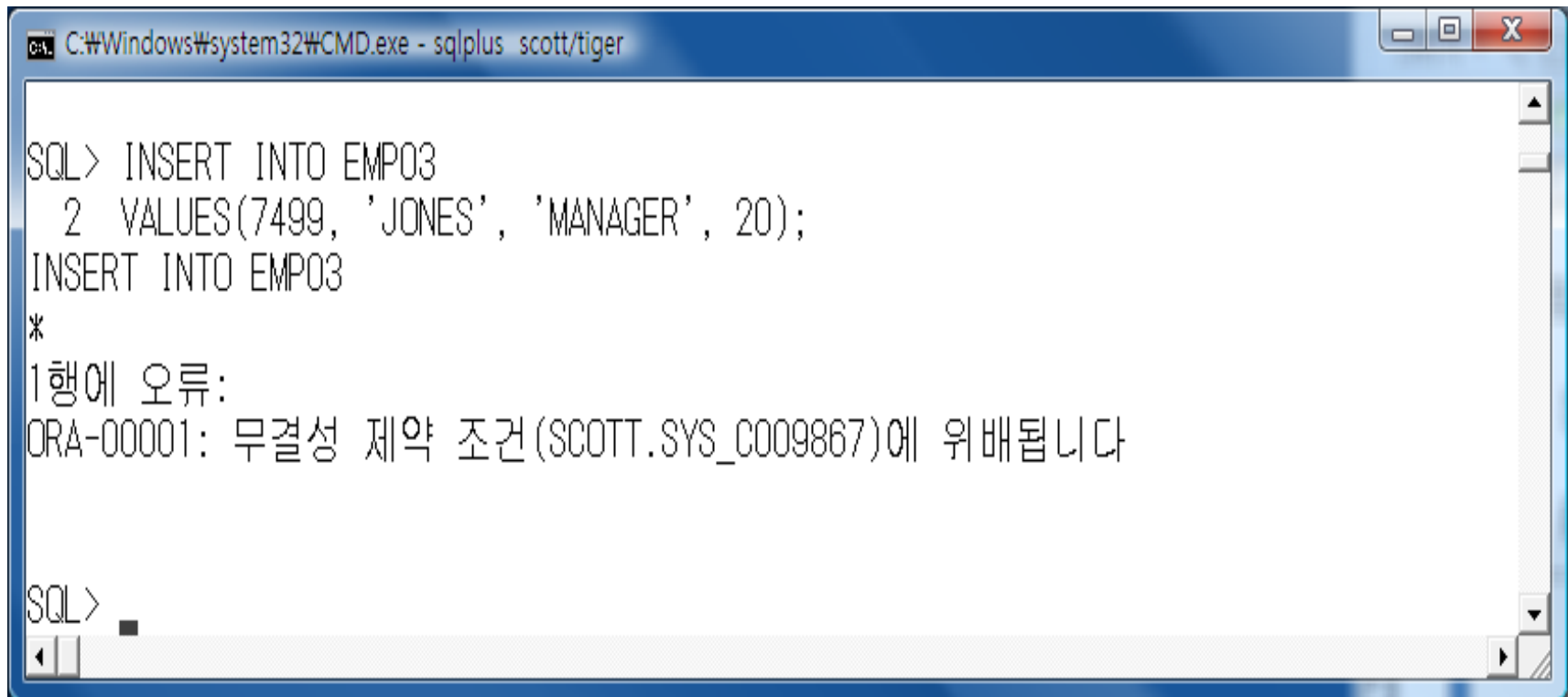
```
C:\Windows\system32\CMD.exe - sqlplus scott/tiger

SQL> INSERT INTO EMP03
  2  VALUES(7499, 'JONES', 'MANAGER', 20);
INSERT INTO EMP03
*
1행에 오류:
ORA-00001: 무결성 제약 조건(SCOTT.SYS_C009867)에 위배됩니다

SQL>
```

제약 조건명을 명시해서 제약 조건 설정하기

- 제약 조건에 위배하면 오류 메시지에 제약 조건명만 출력되는데 오라클이 부여한 제약 조건명으로는 어떤 제약 조건을 위반했는지 알 수 없기에 USER_CONSTRAINTS 데이터 딕셔너리를 검색해야만 어떤 제약 조건인지 확인 할 수 있습니다.
- 만일 사용자가 의미 있게 제약 조건명을 명시한다면 제약 조건명만으로도 어떤 제약 조건을 위배했는지 알 수 있게 됩니다.



```
C:\Windows\system32\CMD.exe - sqlplus scott/tiger

SQL> INSERT INTO EMP03
  2  VALUES(7499, 'JONES', 'MANAGER', 20);
INSERT INTO EMP03
*
1행에 오류:
ORA-00001: 무결성 제약 조건 (SCOTT.SYS_C009867)에 위배됩니다

SQL>
```

제약 조건명을 명시해서 제약 조건 설정하기

- 제약 조건명을 지정하는 방법

```
column_name data_type CONSTRAINT  
constraint_name constraint_type
```

- 사용자 제약 조건 명을 설정하기 위해서는 CONSTRAINT라는 키워드와 함께 제약 조건 명을 기술하면 된다는 것을 확인할 수 있습니다.
- 제약 조건 명(constraining_name)은 다음과 같은 명명 규칙을 준수해서 작성하는 것이 좋습니다.

[테이블명]_[칼럼명]_[제약 조건 유형]

실습하기

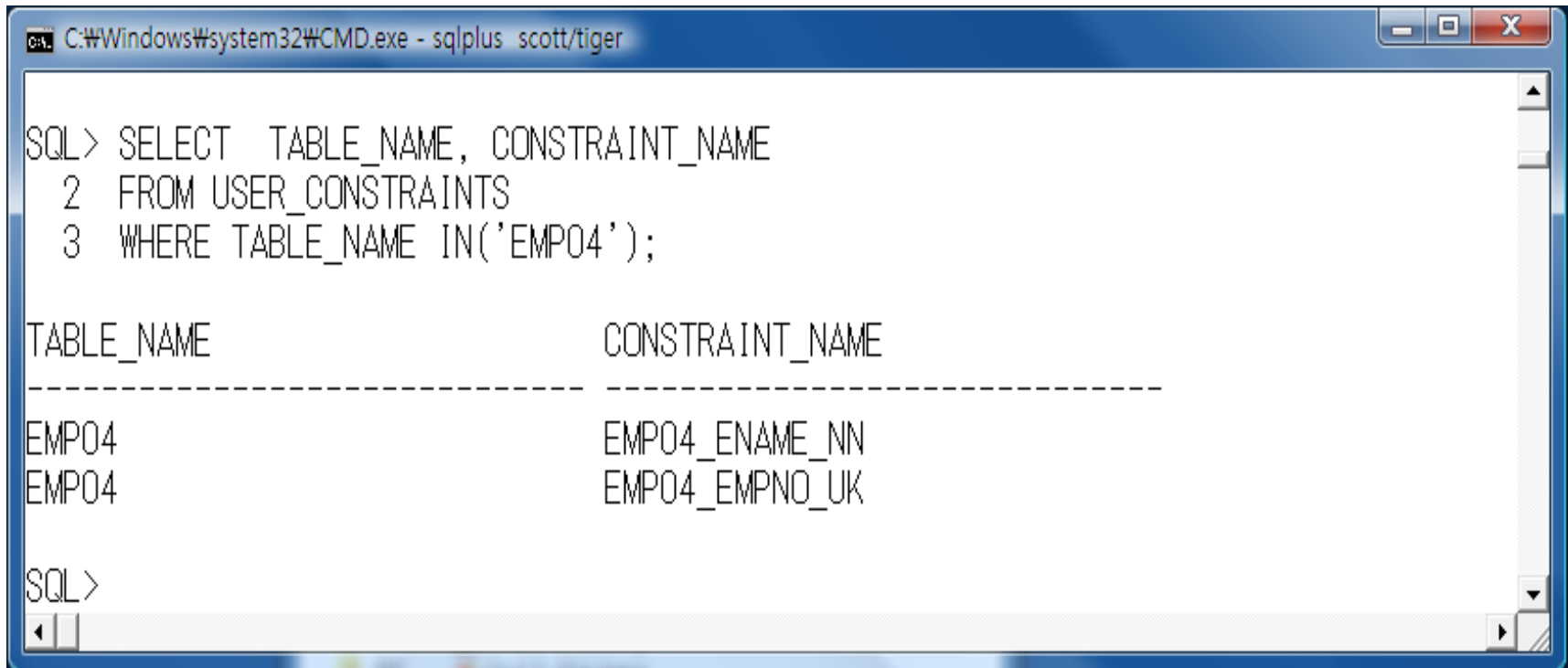
1. 지금까지 실습에 사용했던 사원 테이블과 유사한 구조의 사원번호, 사원명, 직급, 부서번호 4개의 칼럼으로 구성된 EMP04 테이블을 생성하되 사원번호는 중복된 데이터를 삽입할 수 없도록 사원명은 NOT NULL 제약조건을 설정해 봅니다.

```
CREATE TABLE EMP04(  
  EMPNO NUMBER(4) CONSTRAINT EMP04_EMPNO_UK UNIQUE,  
  ENAME VARCHAR2(10) CONSTRAINT EMP04_ENAME_NN NOT  
  NULL,  
  JOB VARCHAR2(9),  
  DEPTNO NUMBER(2)  
);
```

실습하기

2. 생성된 제약 조건 명을 확인하기 위해서 USER_CONSTRAINTS 데이터 디렉터리 뷰를 검색해 봅니다.

```
SELECT TABLE_NAME, CONSTRAINT_NAME  
FROM USER_CONSTRAINTS  
WHERE TABLE_NAME IN('EMP04');
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\CMD.exe - sqlplus scott/tiger". The window contains the following SQL query and its output:

```
SQL> SELECT TABLE_NAME, CONSTRAINT_NAME  
2 FROM USER_CONSTRAINTS  
3 WHERE TABLE_NAME IN('EMP04');
```

TABLE_NAME	CONSTRAINT_NAME
EMP04	EMP04_ENAME_NN
EMP04	EMP04_EMPNO_UK

The output is displayed in a table format with two columns: TABLE_NAME and CONSTRAINT_NAME. The first column lists the table names (EMP04 and EMP04), and the second column lists the constraint names (EMP04_ENAME_NN and EMP04_EMPNO_UK). The prompt "SQL>" is visible at the bottom of the window.

PRIMARY Key 제약조건

- PRIMARY Key 제약조건은 테이블에서 각 행을 유일하게 식별하는 열 또는 열의 집합
- 중복 배제
- 이 제약조건은 열 또는 열의 집합의 유일성을 요구하고 null 값을 포함할 수 없음을 보증함
- 자동으로 Unique INDEX 생성

PRIMARY Key 제약조건

가) PRIMARY KEY(PK)..

- 1) 테이블에 대한 기본 키를 생성합니다...
- 2) 하나의 기본 키만이 각 테이블에 대해 존재할 수 있다...
- 3) PRIMARY KEY 제약 조건은 테이블에서 각행을 유일하게 식별하는 열 또는 열의 집합입니다. (UNIQUE 와 NOT NULL 조건을 만족)..
- 4) 이 제약 조건은 열 또는 열의 집합의 유일성을 요구하고 NULL 값을 포함할 수 없음을 보증 합니다...
- 5) UNIQUE INDEX 가 자동 생성된다...

◆ Syntax..

```
column datatype [CONSTRAINT constraint_name] PRIMARY KEY (col1[,col2,...])..  
column datatype..  
.....  
[CONSTRAINT constraint_name] PRIMARY KEY (column1[,column2,...])..
```

문제 5) 아래의 두 문장의 차이점을 설명하여라..

```
SQL> CREATE TABLE test_tab1(  
2   id          NUMBER(2) CONSTRAINT test_id_pk PRIMARY KEY..  
3   name        VARCHAR2(10));..  
..  
Table created..
```

PRIMARY Key 제약조건

- 열 레벨 또는 테이블 레벨에서 정의될 수 있음
- 조합 PRIMARY Key는 테이블 정의 레벨을 사용하여 생성됨

```
SQL> CREATE TABLE deptpk (  
2         deptno  NUMBER(2),  
3         dname   VARCHAR2(14),  
4         loc     VARCHAR2(13),  
5         CONSTRAINT deptpk_dname_uk UNIQUE(dname),  
6         CONSTRAINT deptpk_deptno_pk PRIMARY Key(deptno));
```

FOREIGN Key 제약조건

- FOREIGN Key, 또는 참조 무결성 제약조건(reference)은 열 또는 열의 집합을 foreign key로 지정하여 동일 테이블 또는 다른 테이블 간의 기본 키 또는 유일 키 사이의 관계를 설정
- 외래 키 값은 부모 테이블에서 존재하는 값과 일치해야 하거나 NULL이 되어야 함
- 외래 키는 데이터 값을 기초로 하여 순전히 논리적이거나 물리적이거나 포인터가 아님

PRIMARY Key 제약조건

```
SQL> CREATE TABLE BUSEO(  
2  B_NO NUMBER(4) CONSTRAINT BUSEO_B_NO_PK PRIMARY KEY,  
3  B_NAME VARCHAR2(10),  
4  B_LOC VARCHAR2(10));
```

테이블이 생성되었습니다.

```
SQL> INSERT INTO BUSEO VALUES (100, '인사과', '서울');
```

1 개의 행이 만들어졌습니다.

```
SQL> INSERT INTO BUSEO VALUES (200, '총무과', '대전');
```

1 개의 행이 만들어졌습니다.

```
SQL> INSERT INTO BUSEO VALUES (300, '경리과', '부산');
```

1 개의 행이 만들어졌습니다.

PRIMARY Key 제약조건

```
SQL> CREATE TABLE SAWON(  
2  S_NO NUMBER(4) CONSTRAINT SAWON_S_NO_PK PRIMARY KEY,  
3  S_NAME VARCHAR2(10),  
4  S_SAL NUMBER(5),  
5  B_NO NUMBER(4) CONSTRAINT SAWON_B_NO_FK REFERENCES BUSEO(B_NO));
```

테이블이 생성되었습니다.

```
SQL> INSERT INTO SAWON VALUES (1, '길동', 1000, 100);
```

1 개의 행이 만들어졌습니다.

```
SQL> INSERT INTO SAWON VALUES (2, '콩쥐', 2000, 150);  
INSERT INTO SAWON VALUES (2, '콩쥐', 2000, 150)
```

*

1행에 오류:

ORA-02291: 무결성 제약조건(SCOTT.SAWON_B_NO_FK)이 위배되었습니다- 부모 키가 없습니다

```
SQL> INSERT INTO SAWON VALUES (3, '팥쥐', 3000, NULL);
```

1 개의 행이 만들어졌습니다.

실습하기

다음은 사원 테이블의 사원번호를 기본 키로 지정한 예입니다. 제약 조건은 칼럼명과 자료형을 기술한 후에 연이어서 PRIMARY KEY를 기술하면 됩니다.

1. 지금까지 실습에 사용했던 사원 테이블과 유사한 구조의 사원번호, 사원명, 직급, 부서번호 4개의 칼럼으로 구성된 테이블을 생성하되 기본 키 제약 조건을 설정해 봅시다.

```
CREATE TABLE EMP05(  
  EMPNO NUMBER(4) CONSTRAINT EMP05_EMPNO_PK PRIMARY KEY ,  
  ENAME VARCHAR2(10) CONSTRAINT EMP05_ENAME_NN NOT NULL ,  
  JOB VARCHAR2(9),  
  DEPTNO NUMBER(2)  
);
```

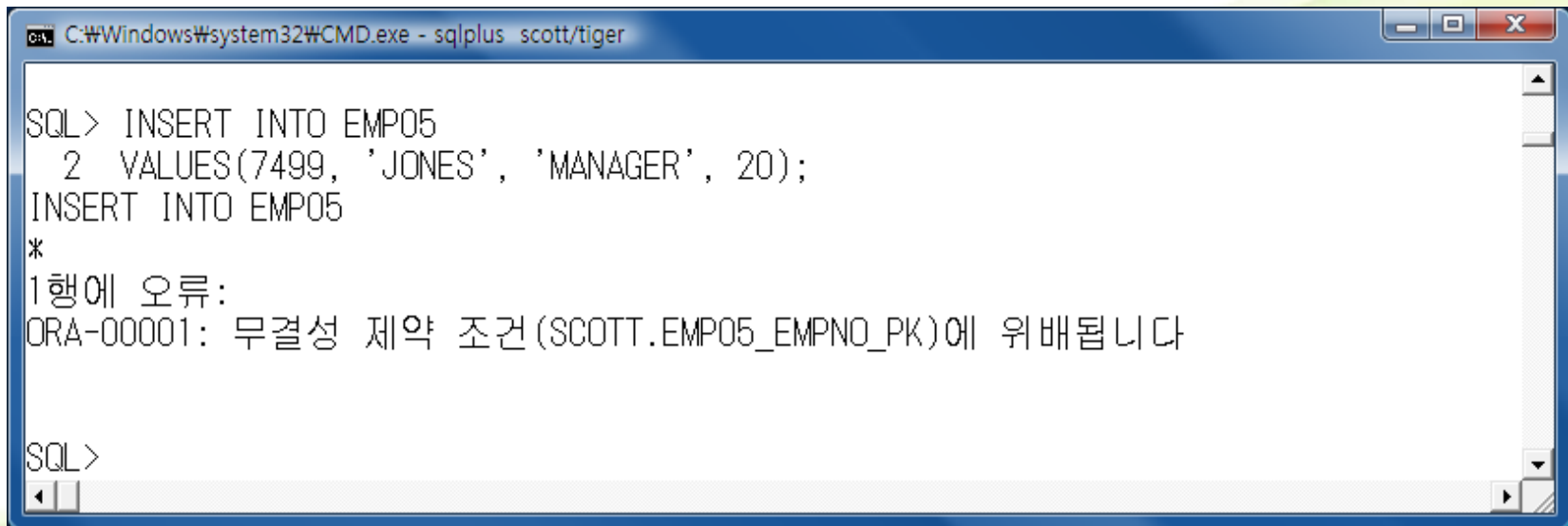
실습하기

2. 위에서 생성한 테이블에 데이터를 추가해 봅시다.

```
INSERT INTO EMP05  
VALUES(7499, 'ALLEN', 'SALESMAN', 30);
```

3. 다음은 기본 키로 지정된 직원번호에 동일한 값을 저장해 보도록 하겠습니다.

```
INSERT INTO EMP05  
VALUES(7499, 'JONES', 'MANAGER', 20);
```



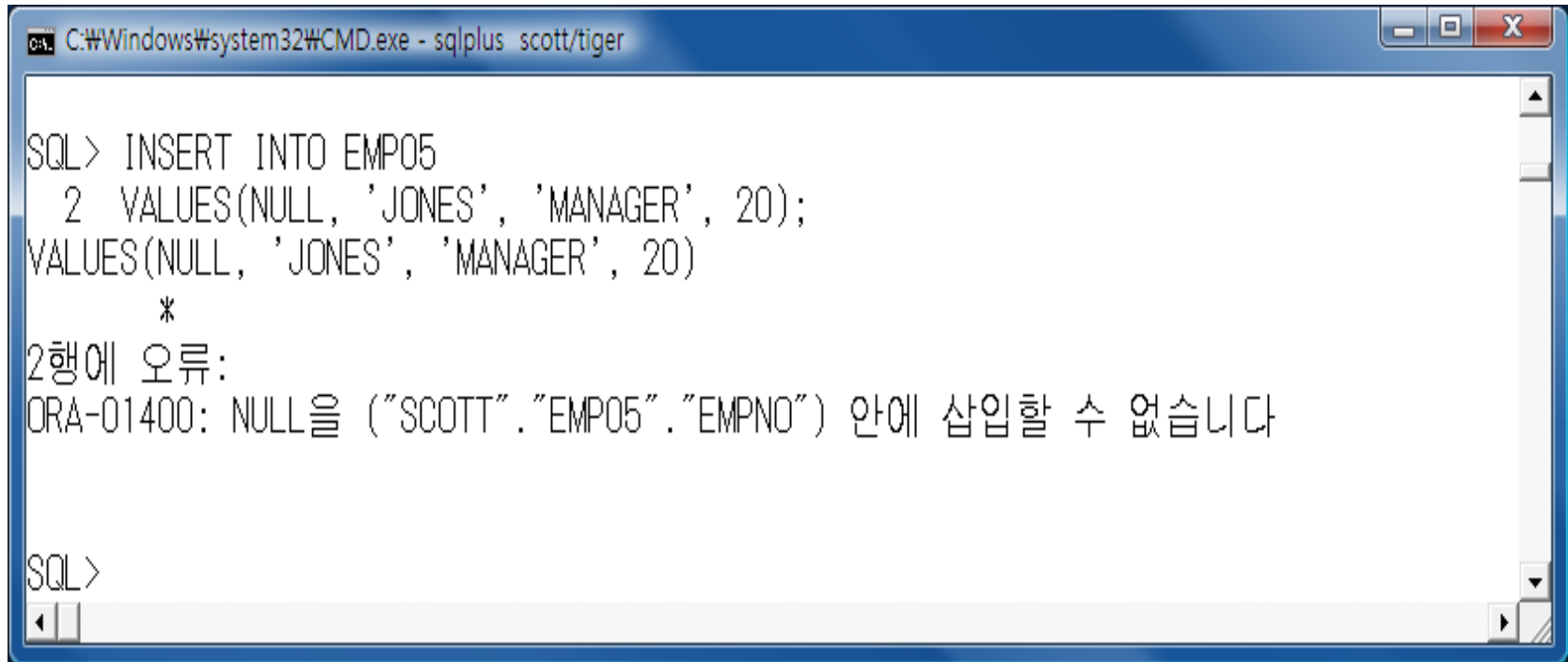
The screenshot shows a Windows command prompt window titled "C:\Windows\system32\CMD.exe - sqlplus scott/tiger". The window contains the following text:

```
SQL> INSERT INTO EMP05  
2  VALUES(7499, 'JONES', 'MANAGER', 20);  
INSERT INTO EMP05  
*  
1행에 오류:  
ORA-00001: 무결성 제약 조건 (SCOTT.EMP05_EMPNO_PK)에 위배됩니다  
  
SQL>
```

실습하기

4. 이번에는 기본 키로 지정된 사원번호에 NULL 값을 저장해 봅니다.

```
INSERT INTO EMP05  
VALUES(NULL, 'JONES', 'MANAGER', 20);
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\CMD.exe - sqlplus scott/tiger". The user has entered the following SQL command:

```
SQL> INSERT INTO EMP05  
2  VALUES(NULL, 'JONES', 'MANAGER', 20);  
VALUES(NULL, 'JONES', 'MANAGER', 20)
```

The command prompt displays an error message:

```
*  
2행에 오류:  
ORA-01400: NULL을 ("SCOTT"."EMP05"."EMPNO") 안에 삽입할 수 없습니다
```

The prompt returns to the SQL> state.

FOREIGN Key 제약조건

- 열 또는 테이블 제약조건 레벨에서 정의될 수 있음
- 조합 외래 키는 테이블 레벨 정의를 사용하여 생성됨

```
SQL> CREATE TABLE empfk (  
2         empno    NUMBER(4),  
3         ename    VARCHAR2(10) NOT NULL,  
4         job      VARCHAR2(9),  
5         mgr      NUMBER(4),  
6         hiredate DATE,  
7         sal      NUMBER(7,2),  
8         comm     NUMBER(7,2),  
9         deptno   NUMBER(7,2),  
10        CONSTRAINT empfk_deptno_fk FOREIGN key (deptno)  
11                REFERENCES deptpk (DEPTNO));
```

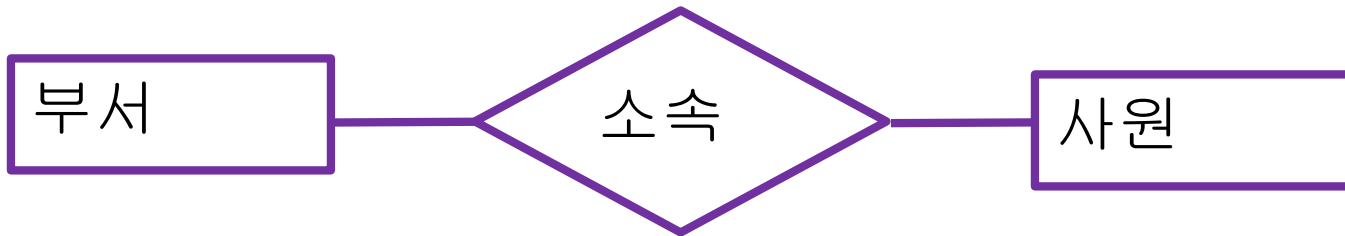
- FOREIGN Key
 - 테이블 제약조건 레벨에서 자식 테이블 열을 정의
- REFERENCES
 - 부모 테이블에서 테이블과 열을 식별함

참조 무결성 (Reference) 제약 조건

- 다른 테이블이나 자신 테이블의 기본키나 고유키를 참조할때 사용
- 참조하는 테이블을 자식테이블, 참조되는 테이블을 부모테이블이라함
- 자식 테이블에서 참조하는 컬럼의 값을 외래키, 부모테이블에서 참조되는 컬럼을 참조키라고함
- 부모 테이블의 컬럼에 존재하지 않는 값을 자식 테이블에 입력할때 오류발생
- 자식 테이블의 값은 부모 테이블의 컬럼에 존재하는 값이거나 널 값
- **ON DELETE CASCADE** 조건은 부모 테이블의 행이 제거될때 자식 테이블에 있는 종속적인 행도 제거됨

FOREIGN Key 제약조건

- 다음은 ERD(Entity Relation Diagram)로서 테이블을 생성하기에 앞서 데이터베이스 모델링 과정에서 업무를 분석한 후 얻어낸 개체와 관계를 다이어그램으로 나타낸 것입니다.



- ERD를 보고 데이터베이스를 구현할 때에는 부서나 사원과 같은 개체는 테이블로 정의하고 소속이란 관계는 참조의 무결성을 위한 특정 컬럼에 외래 키 제약 조건으로 정의합니다.

FOREIGN Key 제약조건

- 참조 무결성은 두 테이블 사이(사원 테이블, 부서 테이블)의 주종 관계에 의해서 결정되는데 주체가 되는 테이블은 부모 테이블이 되고 종속이 되는 테이블은 자식 테이블이 됩니다.

사원은 회사 내에 존재하는 부서에 소속되어 있어야 합니다.

- 사원과 부서의 소속 관계가 위와 같이 표현된다면 부서가 주체(부모 테이블)이고 사원이 종속(자식 테이블)이 됩니다.
- 주체 관계가 애매모호한 경우에는 어느 테이블의 데이터가 먼저 정의되어야 하는가를 기준으로 부모 테이블과 자식 테이블을 구분할 수 있습니다.
- 먼저 정의되어야 하는 테이블이 부모 테이블이고 나중에 정의되어야 하는 테이블이 자식 테이블이 됩니다.
- 회사를 설립하고 어떤 부서를 구성하여 운영할지 정한 후에, 그 부서에서 일할 사원을 뽑아야 소속이란 관계가 성립되므로 부서가 부모 테이블이 되고 사원이 자식 테이블이 됩니다.
- 외래 키(FOREIGN KEY) 제약 조건은 자식 테이블인 사원 테이블(EMP)의 부서번호(DEPTNO) 칼럼에 부모 테이블인 부서 테이블(DEPT)의 부서번호(DEPTNO)를 부모 키로 설정하는 것입니다.

FOREIGN Key 제약조건

- 이때 주의할 점은 부모 키가 되기 위한 칼럼은 반드시 부모 테이블의 기본 키(PRIMARY KEY)나 유일키(UNIQUE)로 설정되어 있어야 한다는 점입니다.

부모테이블

Primary Key		

자식테이블

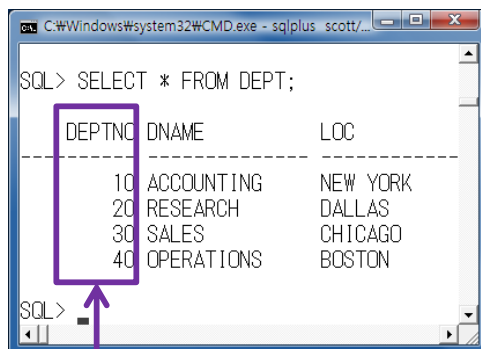
			Foreign Key



- EMP 테이블과 DEPT 테이블을 보면 부모 테이블인 부서 테이블(DEPT)의 부서번호(DEPTNO)는 기본 키(PRIMARY KEY)로 설정되어 있고, 이를 참조할 수 있도록 하기 위해서 자식 테이블인 사원 테이블(EMP)에서 부서번호(DEPTNO)에 외래 키(FOREIGN KEY) 제약조건을 설정해 놓은 상태입니다.

FOREIGN Key 제약조건

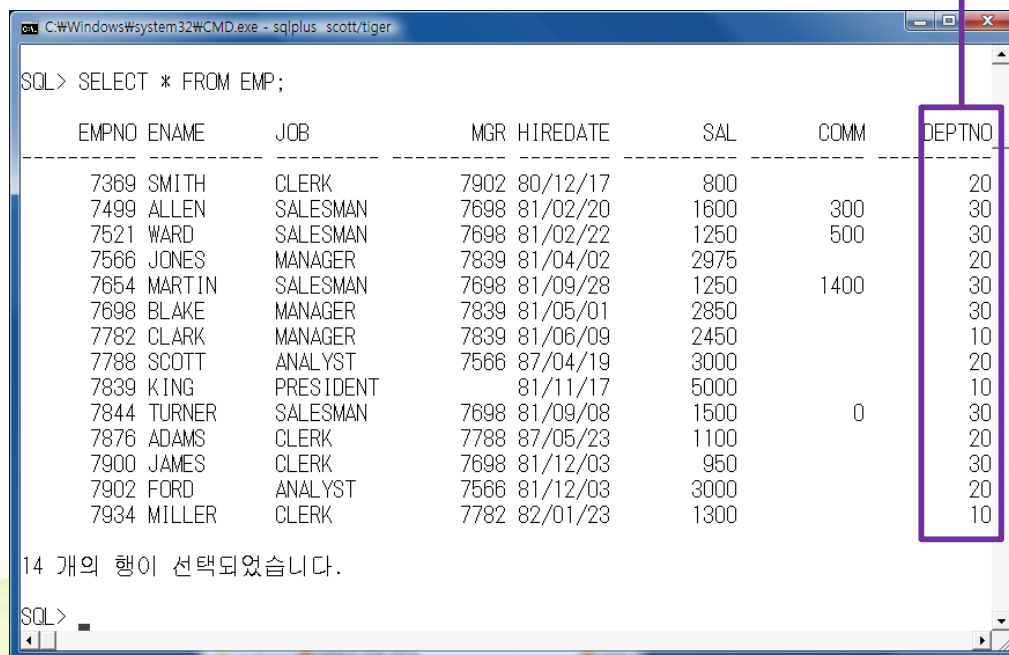
부서(dept) 테이블의 기본 키인 부서 번호 (deptno) 컬럼을 부모 키라고 함



```
SQL> SELECT * FROM DEPT;
```

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

```
SQL>
```



```
SQL> SELECT * FROM EMP;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	80/12/17	800		20
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30
7566	JONES	MANAGER	7839	81/04/02	2975		20
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30
7698	BLAKE	MANAGER	7839	81/05/01	2850		30
7782	CLARK	MANAGER	7839	81/06/09	2450		10
7788	SCOTT	ANALYST	7566	87/04/19	3000		20
7839	KING	PRESIDENT		81/11/17	5000		10
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30
7876	ADAMS	CLERK	7788	87/05/23	1100		20
7900	JAMES	CLERK	7698	81/12/03	950		30
7902	FORD	ANALYST	7566	81/12/03	3000		20
7934	MILLER	CLERK	7782	82/01/23	1300		10

14 개의 행이 선택되었습니다.

```
SQL>
```

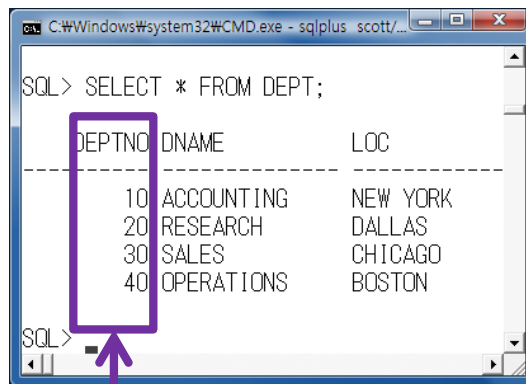
사원(emp) 테이블의 부서 번호 (deptno) 컬럼은 외래 키로 지정해야만 참조의 무결성이 설정됨

FOREIGN Key 제약조건

- 자식 테이블(EMP)에 참조 무결성을 위해 특정 컬럼에 외래 키를 설정하였다면 새로운 데이터를 추가할 때마다 부모 테이블에 부모 키로 설정된 컬럼을 살핍니다.
- 부모 키로 설정된 컬럼에 존재하는 값만 추가하고 존재하지 않는 값이라면 추가하지 않습니다.
- 이렇게 함으로서 자식 테이블이 부모 테이블을 참조하는데 아무런 문제가 없도록 합니다.

FOREIGN Key 제약조건

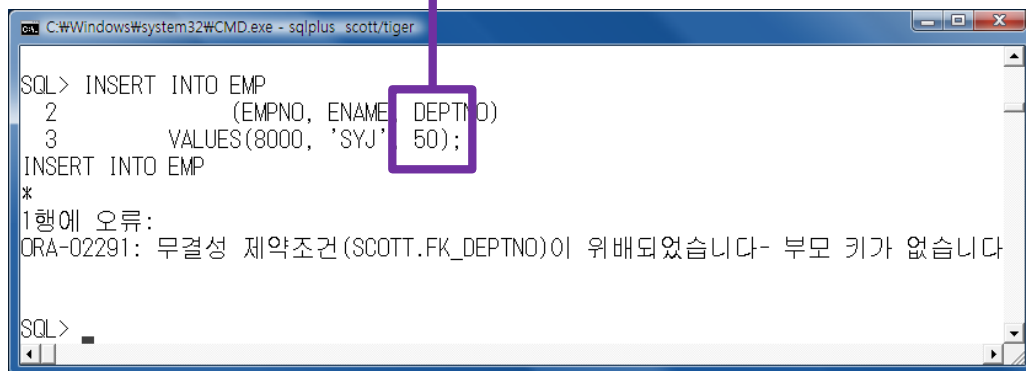
- 외래 키 제약 조건이 지정된 사원 테이블에 부서 테이블에 존재하지 않은 50번 부서번호를 저장해 보도록 합시다.



```
SQL> SELECT * FROM DEPT;
```

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

```
SQL>
```



```
SQL> INSERT INTO EMP
2      (EMPNO, ENAME, DEPTNO)
3      VALUES(8000, 'SYJ', 50);
INSERT INTO EMP
*
1행에 오류:
ORA-02291: 무결성 제약조건 (SCOTT.FK_DEPTNO)이 위반되었습니다- 부모 키가 없습니다

SQL>
```

EMP 테이블에 DEPTNO 컬럼 값을 50으로 하여 새로운 사원을 추가하려 하면 참조의 무결성을 위반했다는 오류 메시지가 출력.

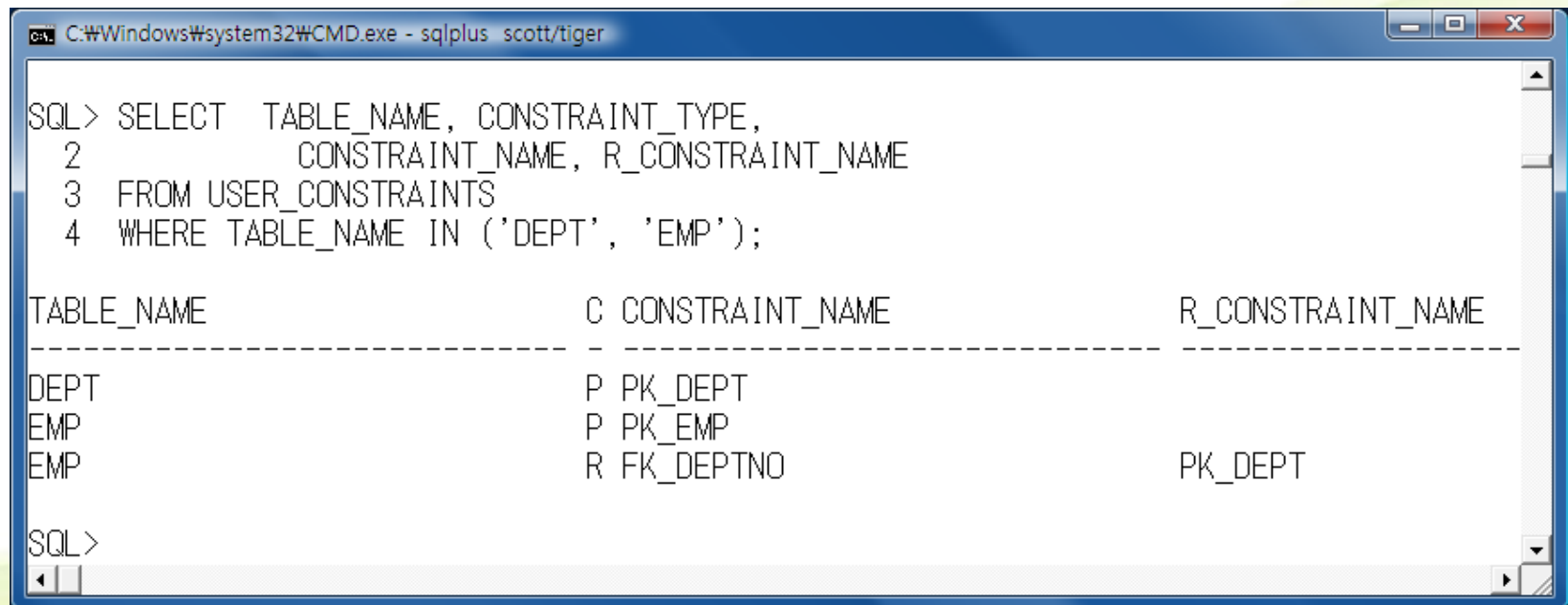
DEPT 테이블에는 DEPTNO 컬럼 값으로 10, 20, 30, 40만 존재하고 50은 존재하지 않기 때문이다.

EMP 테이블에서 참조하는 DEPT 테이블의 DEPTNO 컬럼을 부모 키라고 하므로 부모 키가 없다는 오류 메시지도 함께 출력됨

FOREIGN Key 제약조건

- 다음은 오라클에서 제공해주는 EMP 테이블과 DEPT 테이블의 제약 조건을 살펴보도록 합시다.

```
SELECT TABLE_NAME, CONSTRAINT_TYPE,  
CONSTRAINT_NAME, R_CONSTRAINT_NAME  
FROM USER_CONSTRAINTS  
WHERE TABLE_NAME IN ('DEPT', 'EMP');
```



```
C:\Windows\system32\CMD.exe - sqlplus scott/tiger  
  
SQL> SELECT  TABLE_NAME, CONSTRAINT_TYPE,  
2          CONSTRAINT_NAME, R_CONSTRAINT_NAME  
3 FROM USER_CONSTRAINTS  
4 WHERE TABLE_NAME IN ('DEPT', 'EMP');
```

TABLE_NAME	C CONSTRAINT_NAME	R_CONSTRAINT_NAME
DEPT	P PK_DEPT	
EMP	P PK_EMP	
EMP	R FK_DEPTNO	PK_DEPT

```
SQL>
```

FOREIGN Key 제약조건

- R_CONSTRAINT_NAME 컬럼은 FOREIGN KEY인 경우 어떤 PRIMARY KEY를 참조했는지에 대한 정보를 갖습니다.
- EMP 테이블의 제약조건 FK_DEPTNO의 R_CONSTRAINT_NAME 컬럼값이 PK_DEPT로 설정되어 있습니다.
- 이는 EMP 테이블의 FK_DEPTNO는 외래 키 제약 조건으로 PK_DEPT 제약조건을 참조하고 있다는 내용입니다. PK_DEPT 제약조건은 DEPT 테이블의 기본 키 제약 조건이므로 EMP 테이블은 DEPT 테이블을 참조하고 있는 셈이 됩니다.

실습하기

외래키 제약 조건 설정해 보도록 합시다.

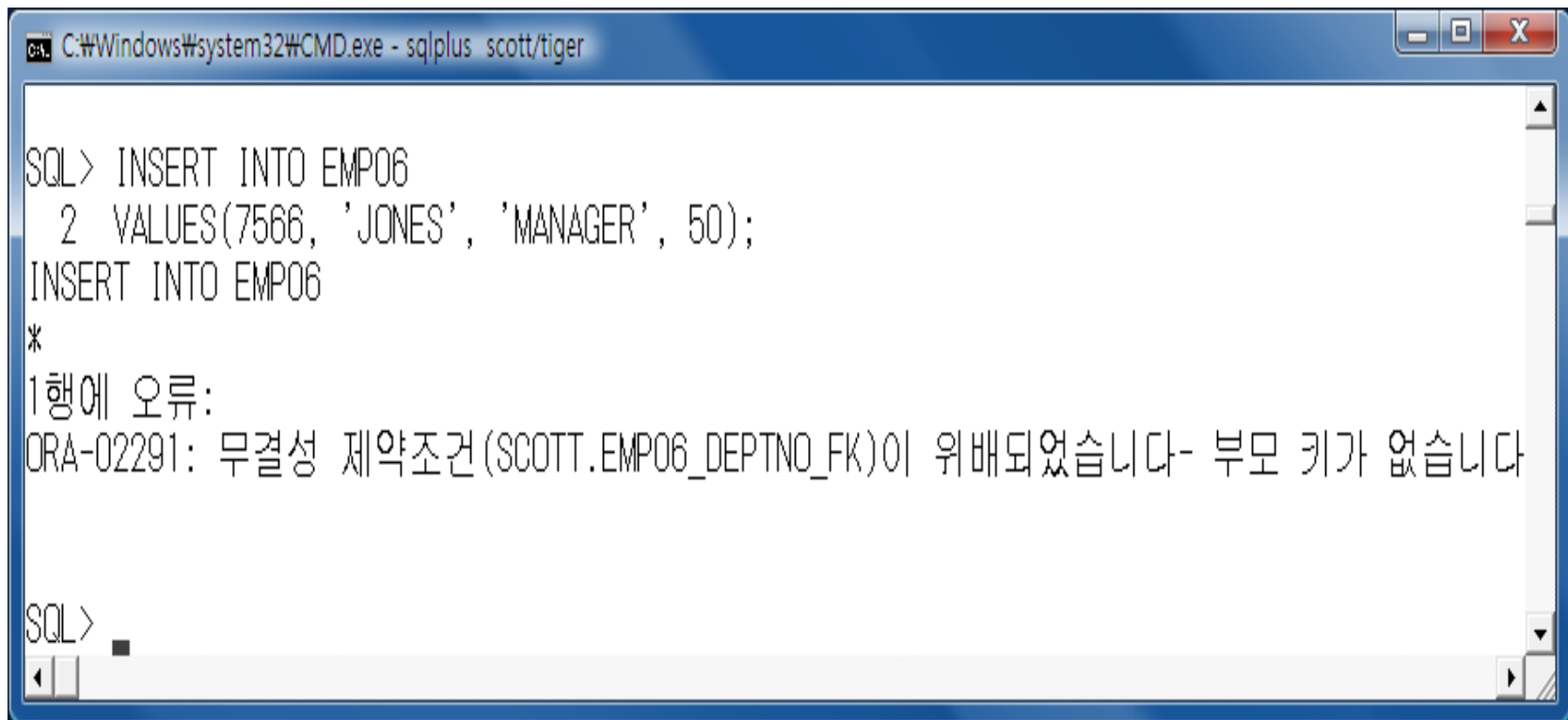
1. 지금까지 실습에 사용했던 사원 테이블과 유사한 구조의 사원번호, 사원명, 직급, 부서번호 4개의 칼럼으로 구성된 테이블을 생성하되 기본 키 제약 조건은 물론 외래키 제약 조건도 설정해 봅시다.

```
CREATE TABLE EMP06(  
  EMPNO NUMBER(4)  
  CONSTRAINT EMP06_EMPNO_PK PRIMARY KEY ,  
  ENAME VARCHAR2(10)  
  CONSTRAINT EMP06_ENAME_NN NOT NULL,  
  JOB VARCHAR2(9),  
  DEPTNO NUMBER(2)  
  CONSTRAINT EMP06_DEPTNO_FK REFERENCES DEPT(DEPTNO)  
);
```

실습하기

2. 현재 EMP06 테이블에 부서 테이블에 존재하지 않는 부서번호를 갖는 사원 정보를 추가해 봅시다.

```
INSERT INTO EMP06  
VALUES(7566, 'JONES', 'MANAGER', 50);
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\CMD.exe - sqlplus scott/tiger". The window contains the following text:

```
SQL> INSERT INTO EMP06  
2  VALUES(7566, 'JONES', 'MANAGER', 50);  
INSERT INTO EMP06  
*  
1행에 오류:  
ORA-02291: 무결성 제약조건 (SCOTT.EMP06_DEPTNO_FK)이 위배되었습니다- 부모 키가 없습니다  
  
SQL>
```

FOREIGN Key 제약조건

- ON DELETE CASCADE를 사용하면 상위 테이블에서 행을 삭제하면 하위 테이블에서 데이터가 자동으로 삭제되도록 할 수 있습니다.

```
column datatype [CONSTRAINT constraint_name]
REFERENCES table_name (column1[,column2,...]
[ON DELETE CASCADE])
```

- ON DELETE SET NULL은 삭제될 때 하위 테이블의 데이터를 null로 변경합니다.
- ON UPDATE CASCADE도 있습니다.

실습하기

외래키 제약 조건 설정해 보도록 합시다.

1. 지금까지 실습에 사용했던 사원 테이블과 유사한 구조의 사원번호, 사원명, 직급, 부서번호 4개의 칼럼으로 구성된 테이블을 생성하되 기본 키 제약 조건은 물론 외래키 제약 조건도 설정해 봅시다.

```
CREATE TABLE EMP06(  
  EMPNO NUMBER(4)  
  CONSTRAINT EMP06_EMPNO_PK PRIMARY KEY ,  
  ENAME VARCHAR2(10)  
  CONSTRAINT EMP06_ENAME_NN NOT NULL,  
  JOB VARCHAR2(9),  
  DEPTNO NUMBER(2)  
  CONSTRAINT EMP06_DEPTNO_FK REFERENCES DEPT(DEPTNO)  
);
```

CHECK 제약 조건

- 각 행이 만족해야 하는 조건을 정의
- 조건은 다음의 경우만을 제외하고 질의 조건과 동일한 구성을 사용할 수 있음
 - CURRVAL, NEXTVAL, LEVEL, ROWNUM, pseudocolumn 에 대한 참조
 - SYSDATE, UID, USER, USERENV 함수에 대한 호출
 - 다른 행에 있는 다른 값을 참조 하는 질의
- 단일 열은 복수 CHECK 제약 조건을 가질 수 없다
- 열에서 정의할 수 있는 CHECK 제약조건의 수에 대한 한계는 없다

CHECK 제약 조건

```
SQL> CREATE TABLE SAWON (  
  2  S_NO NUMBER(4),  
  3  S_NAME VARCHAR2(10),  
  4  S_SAL NUMBER(10) CONSTRAINT SAWON_S_SAL_CHK CHECK (S_SAL > 0));
```

테이블이 생성되었습니다.

```
SQL> INSERT INTO SAWON VALUES (1, '길동', 1000);
```

1 개의 행이 만들어졌습니다.

```
SQL> INSERT INTO SAWON VALUES (2, '콩쥐', -1000);  
INSERT INTO SAWON VALUES (2, '콩쥐', -1000)
```

*

1행에 오류:

ORA-02290: 체크 제약조건(SCOTT.SAWON_S_SAL_CHK)이 위배되었습니다

CHECK 제약 조건

CHECK 제약조건은 해당 컬럼의 값이 반드시 만족해야 될 조건을 지정하는 것으로 제약조건에 CURRVAL, NEXTVAL, LEVEL, ROWNUM과 같은 가상 컬럼(Pseudocolumn)과 SYSDATE, UID, USER, USERENV 함수를 호출 할 수 없으며 다른 행의 값도 참조 할 수 없다. CHECK 제약조건은 컬럼 수준 및 테이블 수준에서 정의 할 수 있다.

```
CREATE TABLE SAWON (  
  S_NO NUMBER(4),  
  S_NAME VARCHAR2(10),  
  S_SAL NUMBER(10),  
  CONSTRAINT SAWON_S_SAL_CHK CHECK (S_SAL > 0));
```

실습하기

사원 테이블에 급여 컬럼을 생성하되 급여 컬럼 값은 500에서 5000사이의 값만 저장할 수 있도록 하고 성별을 저장하는 컬럼으로 GENDER 를 정의하고, 이 컬럼에는 남자는 M, 여자는 F 둘 중의 하나만 저장할 수 있도록 CHECK 제약조건을 지정해 보시다.

1. 사원번호, 사원명, 직급, 부서번호, 직급, 성별 6개의 칼럼으로 구성된 테이블을 생성하되 기본 키 제약 조건, 외래키 제약 조건은 물론 CHECK 제약 조건도 설정해 보시다.

```
CREATE TABLE EMP07(  
  EMPNO NUMBER(4)  
  CONSTRAINT EMP07_EMPNO_PK PRIMARY KEY ,  
  ENAME VARCHAR2(10)  
  CONSTRAINT EMP07_ENAME_NN NOT NULL,  
  SAL NUMBER(7, 2)  
  CONSTRAINT EMP07_SAL_CK CHECK(SAL BETWEEN 500 AND 5000),  
  GENDER VARCHAR2(1)  
  CONSTRAINT EMP07_GENDER_CK CHECK (GENDER IN('M', 'F'))  
);
```

DEFAULT 제약 조건

디폴트는 아무런 값을 입력 하지 않았을 때 디폴트제약의 값이 입력이 됩니다.

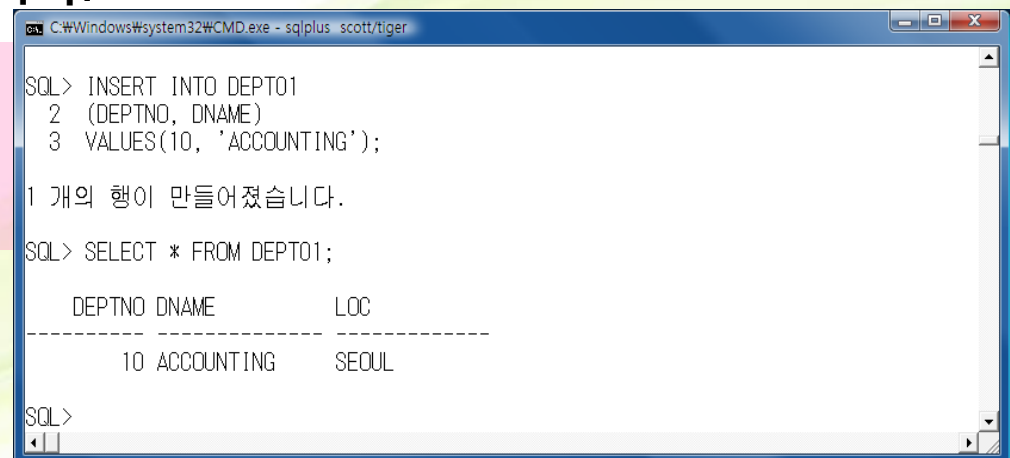
만약 지역명(LOC)라는 컬럼에 아무런 값도 입력 안했을 때 디폴트의 값인 'SEOUL'이 들어가도록 하고 싶을 경우 디폴트 제약 조건을 지정합니다.

다음과 같이 부서 테이블을 생성해 봅시다.

```
CREATE TABLE DEPT01(  
  DEPTNO NUMBER(2) PRIMARY KEY,  
  DNAME VARCHAR2(14),  
  LOC VARCHAR2(13) DEFAULT 'SEOUL'  
);
```

만약 지역명(LOC)라는 컬럼에 아무런 값도 입력하지 않았을 때 디폴트의 값인 'SEOUL'이 들어감을 확인할 수 있습니다.

```
INSERT INTO DEPT01  
(DEPTNO, DNAME)  
VALUES(10, 'ACCOUNTING');
```



```
C:\Windows\system32\CMD.exe - sqlplus scott/tiger  
  
SQL> INSERT INTO DEPT01  
2  (DEPTNO, DNAME)  
3  VALUES(10, 'ACCOUNTING');  
  
1 개의 행이 만들어졌습니다.  
  
SQL> SELECT * FROM DEPT01;  
  
   DEPTNO DNAME      LOC  
-----  
        10 ACCOUNTING  SEOUL  
  
SQL>
```

테이블 레벨 제약 조건

- 지금까지 제약 조건을 지정하는 방식을 컬럼 레벨의 제약 조건 지정이라고 합니다.
- 컬럼 레벨 제약 조건
 - CREATE TABLE로 테이블을 생성하면서 컬럼을 정의하게 되는데 하나의 컬럼 정의가 다 마무리되기 전에 컬럼 명 다음에 타입을 지정하고 그 뒤에 연이어서 제약 조건을 지정하는 방식입니다.
- 테이블 레벨의 제약 조건
 - 컬럼을 모두 정의하고 나서 테이블 정의를 마무리 짓기 전에 따로 생성된 컬럼들에 대한 제약 조건을 한꺼번에 지정하는 것입니다.
- 일반적으로 컬럼 레벨 방식으로 제약조건을 지정하는 것이 훨씬 간편할 텐데 굳이 테이블 레벨의 지정 방식을 사용하는 데에는 2가지 이유가 있습니다.
- 복합키로 기본키를 지정할 경우
 - 지금까지는 한 개의 컬럼으로 기본키를 지정했습니다. 하지만, 경우에 따라서는 2개 이상의 컬럼이 하나의 기본키를 구성하는 경우가 있는데 이를 복합키라고 합니다. 복합키 형태로 제약조건을 지정할 경우에는 컬럼 레벨 형식으로는 불가능하고 반드시 테이블 레벨 방식을 사용해야 합니다.
- ALTER TABLE로 제약 조건을 추가할 때
 - 테이블의 정의가 완료되어서 이미 테이블의 구조가 결정된 후에 나중에 테이블에 제약 조건을 추가하고 할 때에는 테이블 레벨 방식으로 제약 조건을 지정해야 합니다.

테이블 레벨 제약 조건

- 다음은 테이블 레벨 정의 방식의 기본 형식입니다.

```
CREATE TABLE table_name
(column_name1 datatype1,
column_name2 datatype2,
...
[CONSTRAINT constraint_name] constraint_type (column_name)
)
```

- 테이블 레벨에서 칼럼의 제약 조건을 정의할 때 주의할 것은 NOT NULL 조건은 테이블 레벨 정의 방법으로 제약 조건을 지정할 수 없다는 점입니다.

실습하기

컬럼 레벨로 제약 조건과 테이블 레벨로 제약 조건을 지정하는 방법의 차이점을 살펴보기로 합니다.

1. 테이블 레벨로 제약 조건을 지정하기에 앞서 컬럼 레벨로 제약 조건을 지정하는 방법을 살펴보기로 합니다.

```
CREATE TABLE EMP01(  
  EMPNO NUMBER(4) PRIMARY KEY  
  ENAME VARCHAR2(10) NOT NULL,  
  JOB VARCHAR2(9) UNIQUE  
  DEPTNO NUMBER(4) REFERENCES DEPT(DEPTNO)  
);
```

실습하기

2. 위 사용 예는 컬럼 레벨 방식인데, 다음 사용 예인 테이블 레벨 방식과 비교 해보도록 합시다.

```
CREATE TABLE EMP02(  
  EMPNO NUMBER(4),  
  ENAME VARCHAR2(10) NOT NULL,  
  JOB VARCHAR2(9),  
  DEPTNO NUMBER(4),  
  PRIMARY KEY(EMPNO),  
  UNIQUE(JOB),  
  FOREIGN KEY(DEPTNO) REFERENCES DEPT(DEPTNO)  
);
```

실습하기

3. 명시적으로 제약 조건 명을 지정하여 테이블 레벨 방식으로 제약 조건을 지정해 봅시다.

```
CREATE TABLE EMP03(  
  EMPNO NUMBER(4) CONSTRAINT EMP03_ENAME_NN NOT NULL,  
  ENAME VARCHAR2(10),  
  JOB VARCHAR2(9),  
  DEPTNO NUMBER(4),  
  CONSTRAINT EMP03_EMPNO_PK PRIMARY KEY(EMPNO),  
  CONSTRAINT EMP03_JOB_UK UNIQUE(JOB),  
  CONSTRAINT EMP03_DEPTNO_FK FOREIGN KEY(DEPTNO)  
    REFERENCES DEPT(DEPTNO)  
);
```


실습하기

이번에는 복합키를 기본 키로 지정하는 방법을 살펴보도록 합시다.

1. 일련번호를 따로 두지 않고 이름과 핸드폰 번호를 복합하여 이를 기본 키로 지정하기로 합시다.

```
CREATE TABLE MEMBER01(  
  NAME VARCHAR2(10),  
  ADDRESS VARCHAR2(30),  
  HPHONE VARCHAR2(16),  
  CONSTRAINT MEMBER01_COMBO_PK PRIMARY KEY(NAME, HPHONE)  
);
```

제약 조건 추가

ALTER TABLE *table*

ADD [constraint *constraint_name*] type (*column*);

- Add 절을 가지는 ALTER TABLE 문장을 사용하여 기존의 테이블에 대한 제약조건을 추가할 수 있음
- 제약조건을 추가 또는 삭제가능, 수정은 불가능
- 제약조건을 활성화 또는 비활성화
- MODIFY 절을 사용하여 NOT NULL 제약조건을 추가
(Alter Table *table_name* MODIFY (*column_name* NOT NULL))

제약조건 추가

- EMP 테이블 안에 유효한 종업원으로 이미 존재해야 하는 관리자를 나타내는 EMP 테이블에 대하여 FOREIGN key 제약조건을 추가할 수 있다.

```
SQL> ALTER TABLE      emp_test
2      ADD CONSTRAINT emp_mgr_fk
3          FOREIGN key(mgr) REPERENCES emp(empno);
Table altered.
```

실습하기

- 제약조건 없이 테이블을 생성한 후 제약 조건을 추가해 봅시다.
- 1. 지금까지 실습에 사용했던 사원 테이블과 유사한 구조의 사원번호, 사원명, 직급, 부서번호 4개의 칼럼으로 구성된 EMP01 테이블을 제약조건을 하나도 설정하지 않은 채 생성해 봅시다.

```
CREATE TABLE EMP01(  
  EMPNO NUMBER(4),  
  ENAME VARCHAR2(10),  
  JOB VARCHAR2(9),  
  DEPTNO NUMBER(4)  
);
```

실습하기

2. 이제 이미 생성이 완료된 EMP01 테이블에 2가지 제약조건을 설정해 보도록 합시다. 첫 번째는 EMPNO 컬럼에 기본키를 설정하고 두 번째에는 DEPTNO 컬럼에 외래키를 설정해 보도록 합시다.

```
ALTER TABLE EMP01  
ADD CONSTRAINT EMP01_EMPNO_PK PRIMARY KEY(EMPNO);
```

```
ALTER TABLE EMP01  
ADD CONSTRAINT EMP01_DEPTNO_FK  
FOREIGN KEY(DEPTNO) REFERENCES DEPT(DEPTNO);
```

MODIFY로 제약 조건 추가

- NOT NULL 제약 조건을 이미 존재하는 테이블에 추가해 보도록 합시다.
- 이미 존재하는 테이블에 무결성 제약 조건을 추가로 생성하기 위해서 ALTER TABLE ... ADD ... 명령문을 사용하였습니다.
- 하지만 NOT NULL 제약 조건은 ADD 대신 MODIFY 명령문을 사용하므로 사용에 주의해야 합니다.
- 이는 'NULL을 허용하는 상태'에서 'NULL을 허용하지 않는 상태'로 변경하는 것이기 때문입니다.

이미 존재하는 테이블에 NOT NULL 제약 조건을 추가해 보도록 합시다.

1. MODIFY 명령어로 NOT NULL 제약 조건을 설정해 봅시다.

```
ALTER TABLE EMP01  
MODIFY ENAME CONSTRAINT EMP01_ENAME_NN NOT NULL;
```

제약조건 삭제

- 제약조건을 삭제하기 위해서 USER_CONSTRAINTS 와 USER_CONS_COLUMNS 데이터 사전 뷰에서 제약조건 이름을 식별할 수 있음.
- DROP 절과 ALTER TABLE 문장을 사용
- DROP의 CASCADE 옵션은 모든 종속적인 제약조건이 삭제되게 함

```
ALTER TABLE table
DROP { PRIMARY KEY | UNIQUE } (column) |
CONSTRAINT constraint_name [CASCADE];
```

실습하기

사원 테이블에 지정한 제약 조건들을 제거해 보도록 합시다.

1. 기본 키 제약 조건을 제거합니다.

```
ALTER TABLE EMP05  
DROP CONSTRAINT EMP05_EMPNO_PK;
```

2. 사원명에 NULL이 저장될 수 있도록 NOT NULL 제약 조건을 제거해 보도록 합시다.

```
ALTER TABLE EMP05  
DROP CONSTRAINT EMP05_ENAME_NN;
```


제약 조건의 비활성화와 CASCADE

- 제약 조건이 설정되면 항상 그 규칙에 따라 데이터 무결성이 보장됩니다.
- 특별한 업무를 수행하는 과정에서 이러한 제약 조건 때문에 작업이 진행되지 못하는 경우가 생깁니다.
- 그렇다고 제약 조건을 삭제해 버리면 데이터 무결성을 보장받지 못하게 됩니다.
- 그렇기 때문에 오라클에서는 제약 조건을 비활성화시킴으로서 제약 조건을 삭제하지 않고도 제약 조건 사용을 잠시 보류할 수 있는 방법을 제공해줍니다.
- 이렇게 비활성화 된 제약 조건은 원하는 작업을 한 후에는 다시 활성화 상태로 만들어 주어야 합니다.
- 제약 조건을 비활성화, 활성화하는 방법을 살펴보도록 합니다.

실습하기

제약 조건을 비활성화를 위한 실습을 위해서 부서 테이블을 만듭시다. 그런 후에 부서 테이블을 부모 테이블로 하는 사원 테이블을 작성합시다. 그러기 위해서는 부서 테이블의 부서번호가 기본 키로 설정되어 있고, 사원 테이블의 부서번호가 부서 테이블의 부서번호를 참조할 수 있도록 외래 키를 설정해야 합니다.

1.부서 테이블에 부서번호를 기본 키로 지정하여 새로운 테이블을 생성해 봅시다.

```
CREATE TABLE DEPT01(  
  DEPTNO NUMBER(2) CONSTRAINT DEPT01_DEPTNO_PK PRIMARY KEY,  
  DNAME VARCHAR2(14),  
  LOC VARCHAR2(13)  
);
```

실습하기

2. 부서 테이블을 만들었으므로 이제 부서 테이블을 부모 테이블로 하는 사원 테이블을 작성하기 위해서 사원 테이블의 부서번호가 부서 테이블의 부서번호를 참조할 수 있도록 외래 키를 설정합니다.

```
CREATE TABLE EMP01(  
  EMPNO NUMBER(4)  
  CONSTRAINT EMP01_EMPNO_PK PRIMARY KEY ,  
  ENAME VARCHAR2(10)  
  CONSTRAINT EMP01_ENAME_NN NOT NULL,  
  JOB VARCHAR2(9),  
  DEPTNO NUMBER(4)  
  CONSTRAINT EMP01_DEPTNO_FK REFERENCES DEPT01(DEPTNO)  
);
```

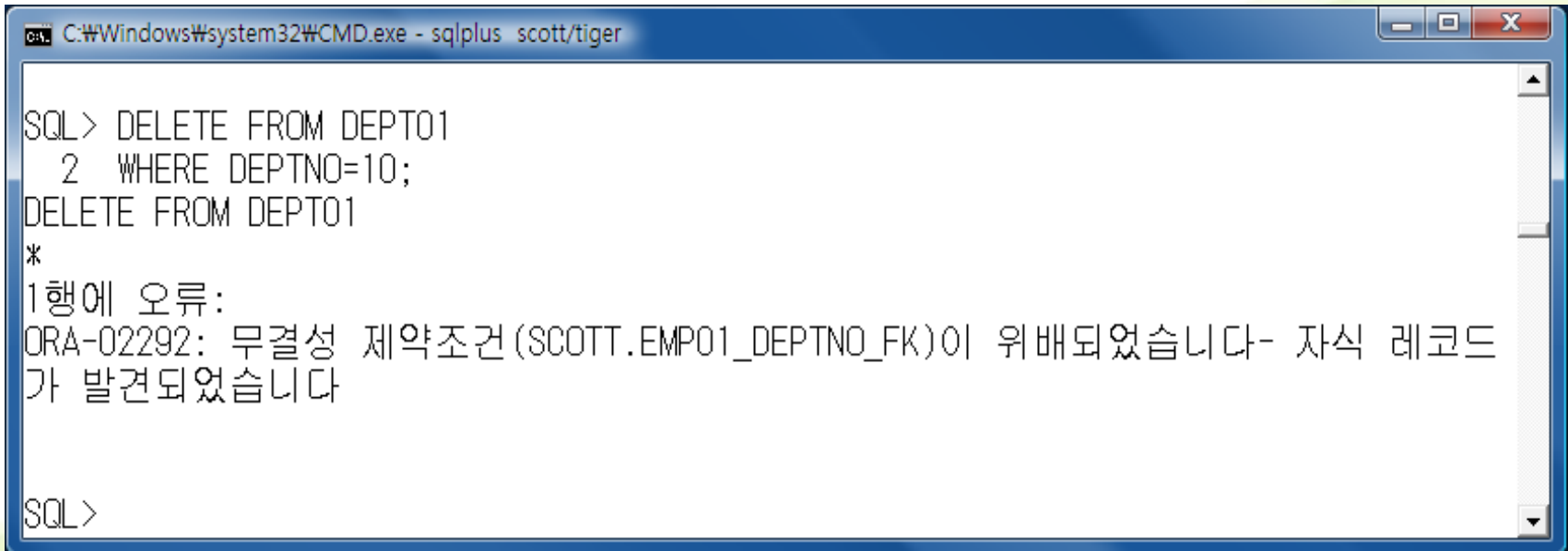
실습하기

3. 사원 테이블로서 사원의 정보를 추가할 때 부서 테이블을 참조하므로 부서 테이블에 존재하는 부서번호를 입력합니다.

```
INSERT INTO EMP01 VALUES(7499, 'ALLEN', 'SALESMAN', 10);  
INSERT INTO EMP01 VALUES(7369, 'SMITH', 'CLERK', 20);
```

4. DEPT01 테이블에서 10번 부서를 ALLEN이란 사람이 참조하고 있는 상태에서 삭제해 봅시다.

```
DELETE FROM DEPT01 WHERE DEPTNO=10;
```



```
C:\Windows\system32\CMD.exe - sqlplus scott/tiger  
  
SQL> DELETE FROM DEPT01  
      2  WHERE DEPTNO=10;  
DELETE FROM DEPT01  
*  
1행에 오류:  
ORA-02292: 무결성 제약조건 (SCOTT.EMP01_DEPTNO_FK)이 위배되었습니다- 자식 레코드가 발견되었습니다  
  
SQL>
```

제약조건 사용불가

- 무결성 제약조건을 비활성화 하기 위해 ALTER TABLE 문장의 DISABLE 절을 실행
- 종속적인 무결성 제약조건을 비 활성화 하기위해서 CASCADE 옵션을 적용함

```
ALTER TABLE table  
DISABLE CONSTRAINT constraint_name [CASCADE];
```

실습하기

EMP01 테이블에 지정한 외래키 제약 조건을 비활성화한 후에 DEPT01 테이블에서 10번 부서를 삭제해 봅시다.

EMP01 테이블에 지정한 외래키 제약 조건을 비활성화 시킵시다

```
ALTER TABLE EMP01  
DISABLE CONSTRAINT EMP01_DEPTNO_FK;
```

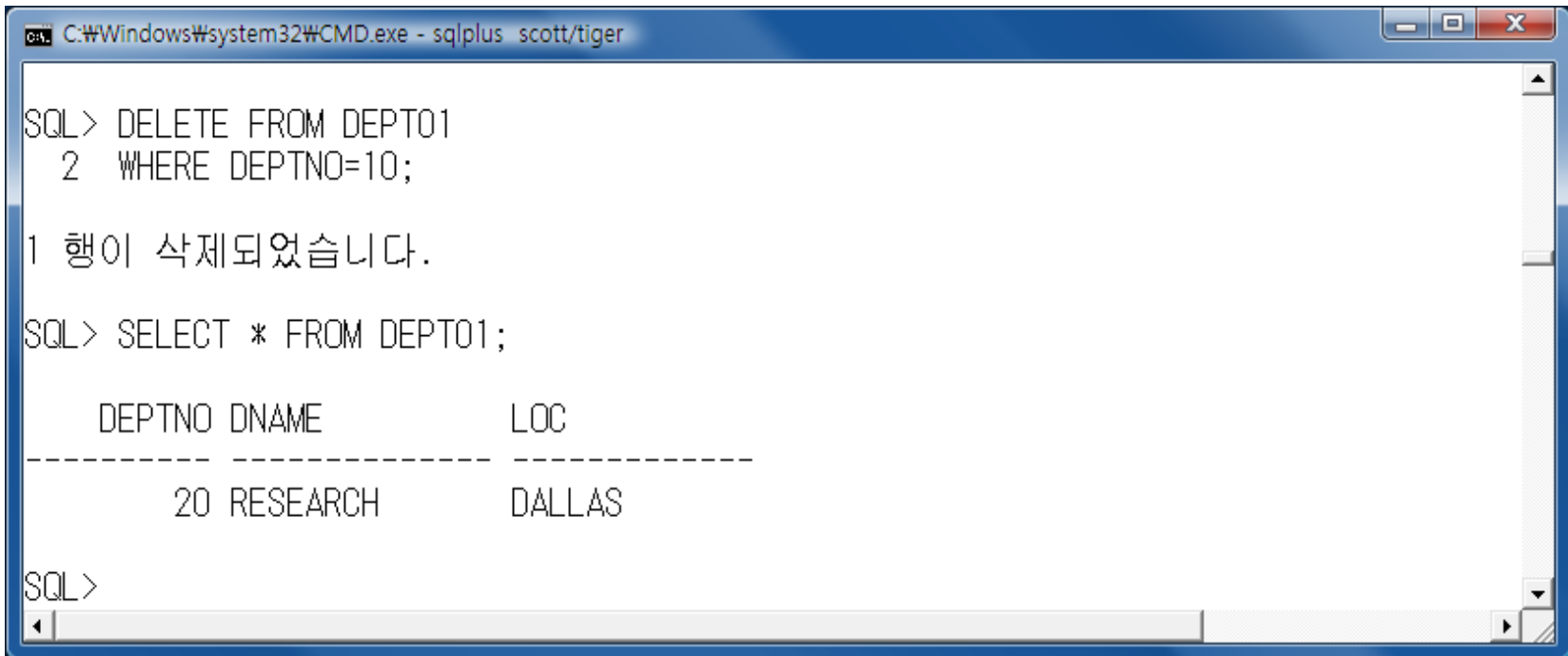
제약 조건의 상태를 확인하기 위해서 USER_CONSTRAINTS 데이터 디렉터리의 STATUS 칼럼값을 살펴보면 EMP01_DEPTNO_FK 제약 조건에 대해서 STATUS 칼럼값이 DISABLED로 지정되어 있음을 확인할 수 있습니다.

```
SELECT CONSTRAINT_NAME, CONSTRAINT_TYPE,  
TABLE_NAME, R_CONSTRAINT_NAME, STATUS  
FROM USER_CONSTRAINTS  
WHERE TABLE_NAME='EMP01';
```

실습하기

2. EMP01 테이블에 지정한 외래키 제약 조건을 비활성화하였기 때문에 DEPT01 테이블에서 10번 부서를 삭제할 수 있게 되었습니다.

```
DELETE FROM DEPT01  
WHERE DEPTNO=10;
```



```
C:\Windows\system32\CMD.exe - sqlplus scott/tiger  
  
SQL> DELETE FROM DEPT01  
2  WHERE DEPTNO=10;  
  
1 행이 삭제되었습니다.  
  
SQL> SELECT * FROM DEPT01;  
  
DEPTNO DNAME      LOC  
-----  
20 RESEARCH      DALLAS  
  
SQL>
```

성공적으로 10번 부서가 삭제되었음을 확인할 수 있습니다. 이는 DEPTNO 칼럼의 FOREIGN KEY 제약 조건이 비활성화되었기 때문에 10번 부서의 자료 삭제가 가능해진 것입니다.

제약조건 사용허가

- ENABLE 절을 가진 ALTER TABLE 문장을 사용하여 삭제 또는 재 생성 없이 제약조건을 활성화 할 수 있음

```
ALTER TABLE table  
ENABLE CONSTRAINT constraint_name [CASCADE];
```

- 제약조건이 활성화 되면 그 제약조건은 테이블의 모든 데이터에 대해 적용됨
- 테이블의 모든 데이터는 제약조건과 일치해야 함
- UNIQUE key 또는 PRIMARY key 제약조건이 활성화 된다면, UNIQUE 또는 PRIMARY key 인덱스는 자동으로 생성됨.
- CREATE TABLE 문장과 ALTER TABLE 문장으로 ENABLE 절을 사용할 수 있다.

실습하기

EMP01 테이블에 지정한 제약 조건 중에서 외래키 제약 조건을 비활성화했습니다. 비활성화된 제약 조건은 다시 활성화해야 합니다.

DISABLE CONSTRAINT 문에 의해 비활성화된 제약 조건을 되살리려면 다음과 같이 ENABLE 을 사용해야 합니다.

```
ALTER TABLE EMP01  
ENABLE CONSTRAINT EMP01_DEPTNO_FK;
```

하지만, 부서 테이블의 10번 부서가 삭제된 상태에서는 외래키 제약 조건을 활성화 시킬 수 없습니다.

왜냐하면 외래키 제약 조건은 참조의 무결성을 위배하지 않은 상태에서만 지정할 수 있는데, 사원 테이블(EMP01)에서 부서 테이블(DEPT01)의 10번 부서를 참조하고 있고 부서 테이블에 10번 부서가 존재하지 않기 때문에 참조의 무결성에 위배되기 때문입니다.

실습하기

2. 그러므로 외래키 제약 조건을 활성화시키기 전에 먼저 삭제된 부서 테이블의 10번 부서를 새로 입력해 놓아야 합니다.

```
INSERT INTO DEPT01  
VALUES(10, 'ACCOUNTING', 'NEW YORK');  
SELECT * FROM DEPT01;
```

3. 10번 부서를 새로 입력해 놓았으므로 이제 외래키 제약 조건을 활성화합니다.

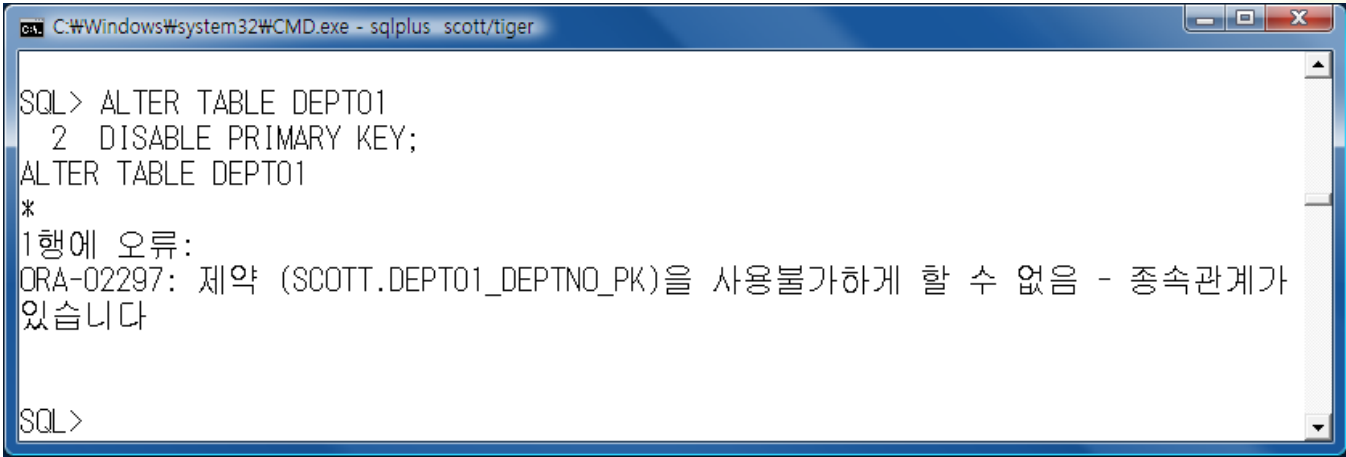
```
ALTER TABLE EMP01  
ENABLE CONSTRAINT EMP01_DEPTNO_FK;
```

CASCADE 옵션

- CASCADE 옵션은 부모 테이블과 자식 테이블 간의 참조 설정이 되어 있을 때 부모 테이블의 제약 조건을 비활성화하면 이를 참조하고 있는 자식 테이블의 제약 조건까지 같이 비활성화시켜 주는 옵션입니다.
- 또한 제약 조건의 비활성화뿐만 아니라 제약 조건이 삭제에도 활용되며, 역시 같은 이치로 부모 테이블의 제약 조건을 삭제하면 이를 참조하고 있는 자식 테이블의 제약 조건도 같이 삭제됩니다.

```
ALTER TABLE DEPT01  
DISABLE PRIMARY KEY;
```

- 부서 테이블(DEPT01)의 기본 키 제약 조건을 비활성화해 보도록 합시다.
1. 부서 테이블(DEPT01)의 기본 키 제약 조건을 "DISABLE PRIMARY KEY"로 비 활성화하려고 시도합니다.



```
C:\Windows\system32\CMD.exe - sqlplus scott/tiger  
  
SQL> ALTER TABLE DEPT01  
2  DISABLE PRIMARY KEY;  
ALTER TABLE DEPT01  
*  
1행에 오류:  
ORA-02297: 제약 (SCOTT.DEPT01_DEPTNO_PK)을 사용불가하게 할 수 없음 - 종속관계가  
있습니다  
  
SQL>
```

- 부서 테이블의 기본 키는 사원 테이블(EMP01)의 외래 키에서 참조하고 있기 때문에 제약 조건을 비활성화할 수 없습니다.

CASCADE 옵션

- 부모 테이블(부서)의 기본 키에 대한 제약조건을 비활성화하고자 하는 것인데 자식 테이블(사원)에서 이를 외래 키 제약조건으로 지정한 컬럼이라면 절대 비활성화할 수 없습니다.
- 만일 비활성화될 수 있다고 가정하면 기본 키가 더 이상 아닌 상태 일반 컬럼을 자식 테이블이 외래 키 제약조건으로 지정하고 있는 아이러니한 상태가 되기 때문입니다.
- 그렇기 때문에 부모 테이블(부서)의 기본 키에 대한 제약조건을 비활성화하려면 자식 테이블(사원)의 외래 키에 대한 제약조건을 비활성화하는 작업이 선행되어야 합니다.
- 두 테이블 사이에 아무런 관련이 없어야 만 즉, 부서 테이블이 더 이상 부모 테이블로서의 역할을 하지 않고 있어야만 기본 키 제약 조건을 비활성화 시킬 수 있습니다.

CASCADE 옵션

- 1) 부모 테이블의 기본 키를 참조하는 자식 테이블의 외래 키에 대한 제약 조건을 비활성화해야 합니다.
 - 2) 부모 테이블의 기본 키에 대한 제약 조건을 비활성화해야 합니다.
- 위 순서대로 제약조건을 여러 번에 걸쳐 비활성화 시키기는 번거로움을 없애주는 것이 CASCADE 옵션입니다.
 - CASCADE 옵션을 지정하여 기본 키 제약 조건을 비활성화하면 이를 참조하는 외래 키 제약 조건도 연속적으로 비활성화되기 때문에 한 번만 비활성화 해 주면 됩니다.

CASCADE 옵션

- 1) 부모 테이블의 기본 키를 참조하는 자식 테이블의 외래 키에 대한 제약 조건을 비활성화해야 합니다.
 - 2) 부모 테이블의 기본 키에 대한 제약 조건을 비활성화해야 합니다.
- 위 순서대로 제약조건을 여러 번에 걸쳐 비활성화 시키기는 번거로움을 없애주는 것이 CASCADE 옵션입니다.
 - CASCADE 옵션을 지정하여 기본 키 제약 조건을 비활성화하면 이를 참조하는 외래 키 제약 조건도 연속적으로 비활성화되기 때문에 한 번만 비활성화 해 주면 됩니다.

실습하기

- CASCADE 옵션을 지정하여 기본 키 제약 조건을 비활성화하면 이를 참조하는 외래 키 제약 조건도 연속적으로 비활성화합니다.
- 1. 부서 테이블(DEPT01)의 기본 키 제약 조건을 CASCADE 옵션을 지정하여 비활성화합니다.

```
ALTER TABLE DEPT01  
DISABLE PRIMARY KEY CASCADE;
```

- 2. 데이터 디렉터리 USER_CONSTRAINTS를 살펴봅시다.

```
SELECT CONSTRAINT_NAME, CONSTRAINT_TYPE,  
TABLE_NAME, R_CONSTRAINT_NAME, STATUS  
FROM USER_CONSTRAINTS  
WHERE TABLE_NAME IN('DEPT01', 'EMP01');
```

실습하기

다음 ERD를 보고 테이블을 작성하시오. 단, 제약조건은 PRIMARY KEY, FOREIGN KEY만 사용하시오.

