

제약조건

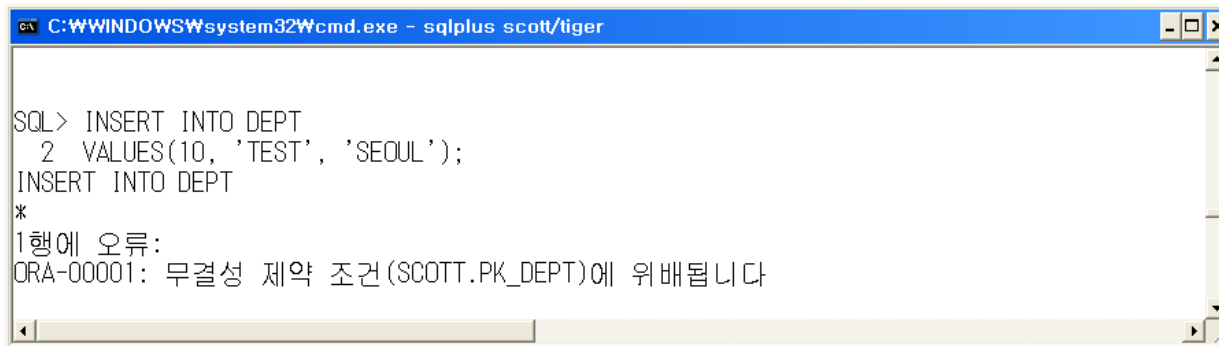
1. 제약조건

- ❖ 데이터 무결성 제약조건(Data Integrity Constraint Rule)이란 테이블에 부적절한 자료가 입력되는 것을 방지하기 위해서 테이블을 생성할 때 각 컬럼에 대해서 정의하는 여러 가지 규칙

무결성 제약조건	역할
NOT NULL	NULL을 허용하지 않음
UNIQUE	중복된 값을 허용하지 않음 항상 유일한 값
PRIMARY KEY	NULL을 허용하지 않고 중복된 값을 허용하지 않음 NOT NULL 조건과 UNIQUE 조건을 결합한 형태로 테이블에 1개만 설정
FOREIGN KEY	참조되는 테이블의 칼럼의 값이 존재하는 경우와 NULL 만 허용
CHECK	저장 가능한 데이터 값의 범위나 조건을 지정하여 설정한 값만을 허용

2. 제약조건 확인

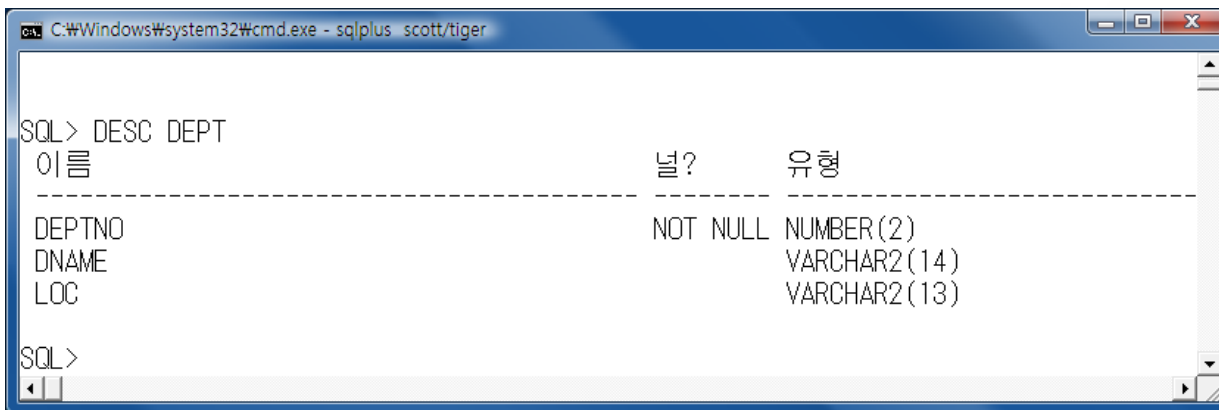
- ❖ 아래의 그림은 EMP 테이블에 INSERT 작업 중 무결성 제약조건을 위반했을 때 나타나는 에러 메시지



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

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

- ❖ DESC 명령어로는 NOT NULL 제약조건만 확인할 수 있고 DEPTNO 컬럼에 기본 키 제약조건이 지정된 것을 알 수 없음



```
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>
```

2. 제약조건 확인

- ❖ 오라클은 USER_CONSTRAINTS 데이터 디렉터리 뷰로 제약조건에 관한 정보를 저장
- ❖ USER_CONSTRAINTS 데이터 디렉터리 뷰를 조회하면 내가 만든(USER) 제약조건(CONSTRAINTS)의 정보를 조회 할 수 있음
- ❖ 테이블의 구조확인

DESC USER_CONSTRAINTS;



2. 제약조건 확인

제약조건을
소유한 사용자명

제약조건 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)

제약조건이 속한 테이블명

FOREIGN KEY인 경우
어떤 PRIMARY KEY를
참조 했는지에 대한 정
보를 갖음

2. 제약조건 확인

- ❖ USER_CONSTRAINTS 데이터 딕셔너리는 제약조건의 정보를 위해서 많은 칼럼으로 구성
- ❖ OWNER는 제약조건을 소유한 사용자이름을 저장하는 컬럼
- ❖ CONSTRAINT_NAME은 제약조건 명
- ❖ CONSTRAINT_TYPE는 제약조건 유형을 저장하는 컬럼
 - ✓ CONSTRAINT_TYPE은 P, R, U, C 4가지 값 중에 하나를 저장

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

2. 제약조건 확인

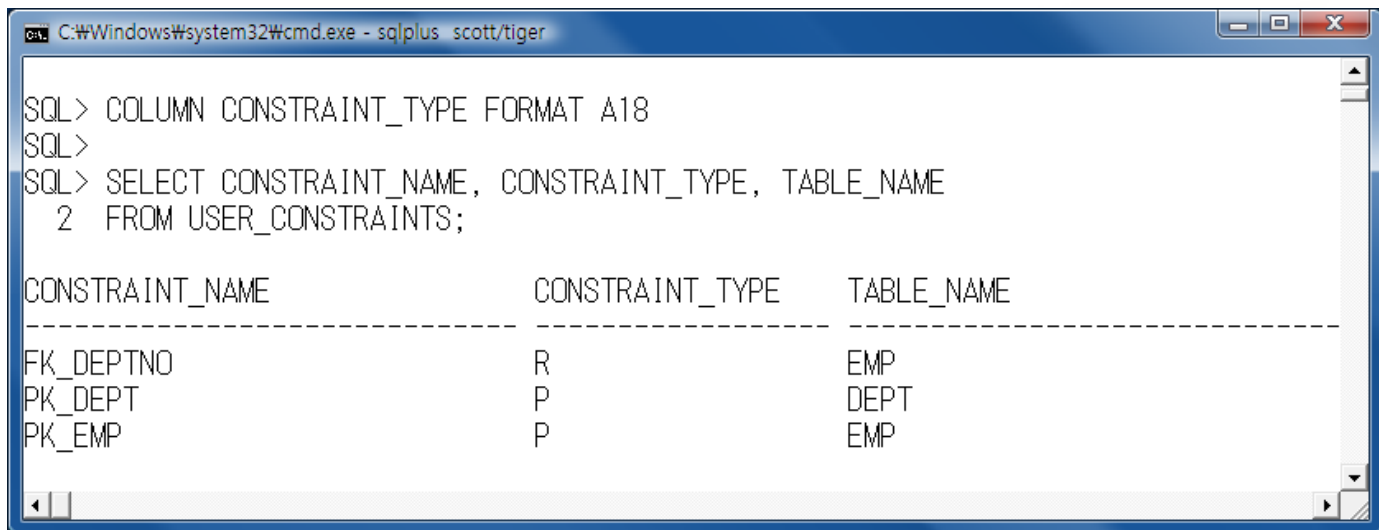
- ❖ TABLE_NAME은 각 제약조건들이 속한 테이블의 이름
- ❖ 제약조건 유형은 4가지
 - ✓ CHECK
 - 제약조건 중에 NOT NULL 은 컬럼에 NULL 값이 저장되어서는 안 된다는 조건
 - NOT NULL 조건은 컬럼에 NULL 값을 체크하는 조건으로 처리되기 때문에 CHECK 를 나타내는 C로 표현
 - C는 CHECK 조건과 NOT NULL 조건을 모두 포함
 - 제약조건 유형이 C인 경우에는 NOT NULL 조건과 CHECK 조건이 모두 포함되는데 NOT NULL 조건인 경우에는 따로 언급되는 내용이 없지만 CHECK 조건일 경우에는 어떤 내용이 조건으로 사용되었는지를 기록하고 있어야 하는데 SEARCH_CONDITION은 제약조건 유형이 C인 경우 각 행에 대한 조건을 설명
 - ✓ FOREIGN KEY
 - FOREIGN KEY는 PRIMARY KEY를 참조하기 때문에 참조 무결성을 지켜야 함
 - 참조 무결성(REFERENCE INTEGRITY)의 이니셜인 R을 FOREIGN KEY 의 제약조건 유형으로 사용
 - R_CONSTRAINT_NAME은 제약조건이 FOREIGN KEY인 경우 어떤 PRIMARY KEY를 참조했는지에 대한 정보를 저장

2. 제약조건 확인

- ❖ 테이블에 어떤 제약조건들이 사용되었는지 살펴보기 위해 Data Dictionary 중 USER_CONSTRAINTS의 내용을 출력

```
SELECT CONSTRAINT_NAME, CONSTRAINT_TYPE, TABLE_NAME
        ①           ②           ③
FROM USER_CONSTRAINTS;
        ④
```

- ❖ ④의 USER_CONSTRAINTS는 제약조건에 대한 모든 정보가 저장된 데이터 디렉터리
- ❖ ①, ②, ③은 데이터디렉터리인 USER_CONSTRAINTS 테이블 내의 컬럼인데 ①은 제약조건 이름을 ②는 제약조건 유형을 ③은 제약조건이 어느 테이블 소속인지 테이블 이름을 저장하고 있는 컬럼



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

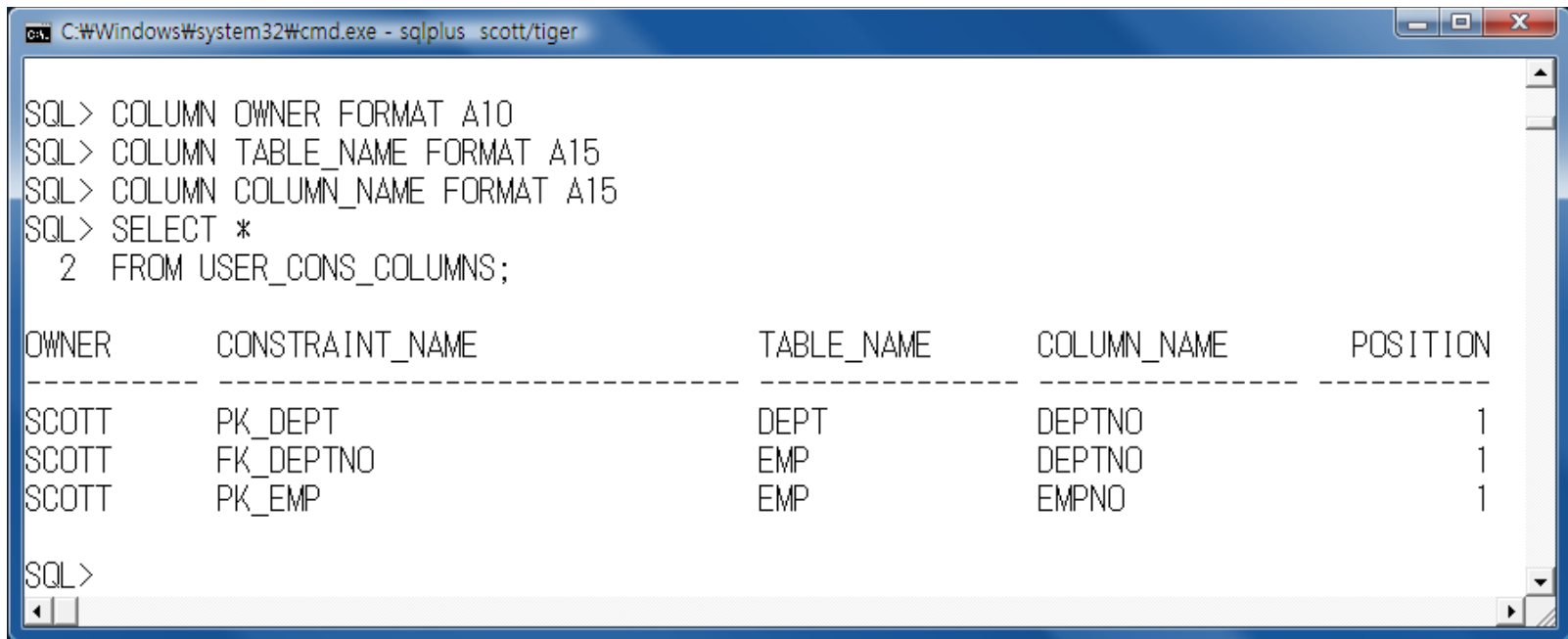
SQL> COLUMN CONSTRAINT_TYPE FORMAT A18
SQL>
SQL> SELECT CONSTRAINT_NAME, CONSTRAINT_TYPE, TABLE_NAME
        2 FROM USER_CONSTRAINTS;
```

CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME
FK_DEPTNO	R	EMP
PK_DEPT	P	DEPT
PK_EMP	P	EMP

2. 제약조건 확인

- ❖ USER_CONSTRAINTS 데이터 디렉터리 뷰에는 어떤 칼럼에 제약조건이 정의되었는지 칼럼 명이 보이지 않음
- ❖ USER_CONS_COLUMNS 데이터 디렉터리 뷰는 제약조건이 지정된 칼럼 명도 알려줌

```
SELECT *  
FROM USER_CONS_COLUMNS;
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". Inside the window, the following SQL commands are entered and executed:

```
SQL> COLUMN OWNER FORMAT A10  
SQL> COLUMN TABLE_NAME FORMAT A15  
SQL> COLUMN COLUMN_NAME FORMAT A15  
SQL> SELECT *  
2 FROM USER_CONS_COLUMNS;
```

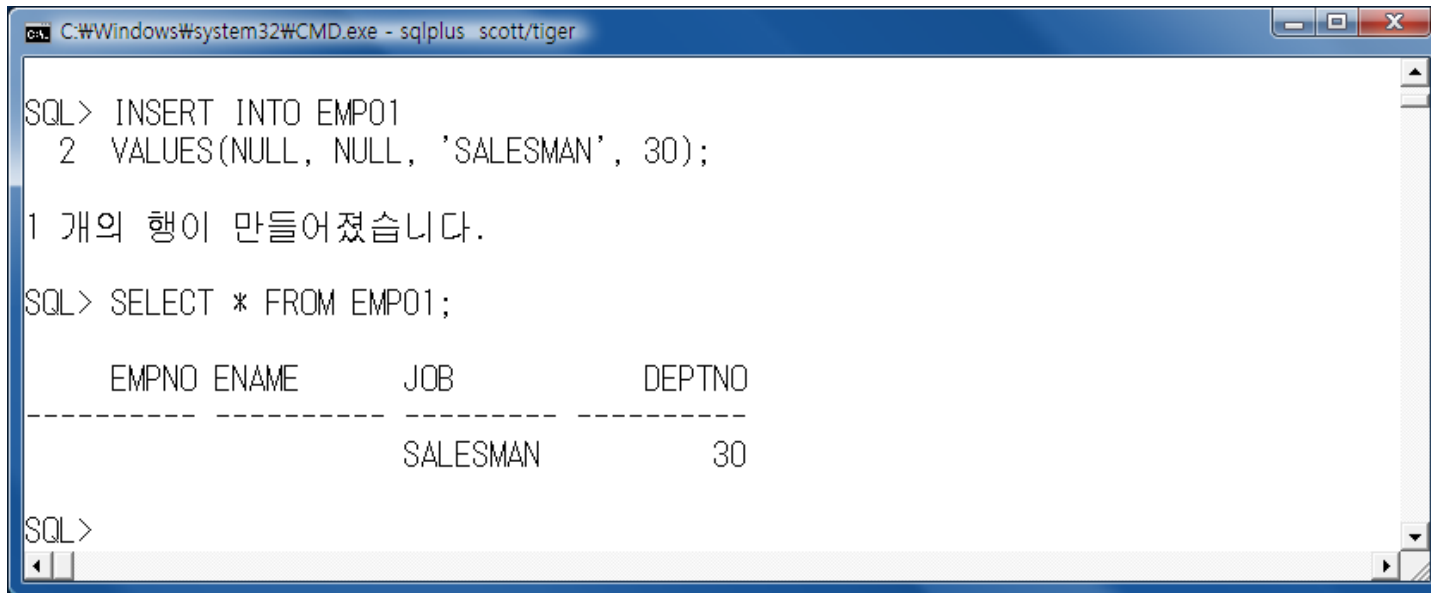
The output of the query is displayed as a table with the following columns: OWNER, CONSTRAINT_NAME, TABLE_NAME, COLUMN_NAME, and POSITION. The data is as follows:

OWNER	CONSTRAINT_NAME	TABLE_NAME	COLUMN_NAME	POSITION
SCOTT	PK_DEPT	DEPT	DEPTNO	1
SCOTT	FK_DEPTNO	EMP	DEPTNO	1
SCOTT	PK_EMP	EMP	EMPNO	1

The command prompt shows the prompt "SQL>" at the bottom, indicating the query has been executed.

3. NOT NULL 제약조건

- ❖ 새로운 사원이 입사하여 사원의 정보를 입력하는데 사원번호와 사원 명이 불 분명하여 데이터가 저장되지 않았다면 누구의 직급인지 누구의 부서번호인지를 모르게 되므로 자료로서의 의미를 갖기 어려움



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

SQL> INSERT INTO EMP01
  2  VALUES(NULL, NULL, 'SALESMAN', 30);

1 개의 행이 만들어졌습니다.

SQL> SELECT * FROM EMP01;
```

EMPNO	ENAME	JOB	DEPTNO
		SALESMAN	30

```
SQL>
```

- ❖ 사원의 정보를 입력할 때 반드시 입력해야 하는 선택이 아닌 필수 입력을 요구하는 컬럼이 있다면 위와 같이 NULL 값이 저장되지 못하도록 제약조건을 설정해야 함
- ❖ NOT NULL 제약조건은 해당 컬럼에 데이터를 추가하거나 수정할 때 NULL 값이 저장되지 않게 제약을 설정하는 것으로 사원번호와 사원명과 같이 자료가 꼭 입력되게 하고 싶을 때 사용

실습하기

- ❖ 사원 테이블과 유사한 구조의 사원번호, 사원명, 직급, 부서번호 4개의 칼럼으로 구성된 EMP01 테이블에 아무런 제약조건을 설정하지 않고 생성한 후에 이렇게 생성한 테이블에는 NULL 값이 저장 가능
- 1. 사원 테이블과 유사한 구조의 사원번호, 사원명, 직급, 부서번호 4개의 칼럼으로 구성된 EMP01 테이블을 생성

```
CREATE TABLE EMP01(  
    EMPNO NUMBER(4),  
    ENAME VARCHAR2(10),  
    JOB VARCHAR2(9),  
    DEPTNO NUMBER(2));
```

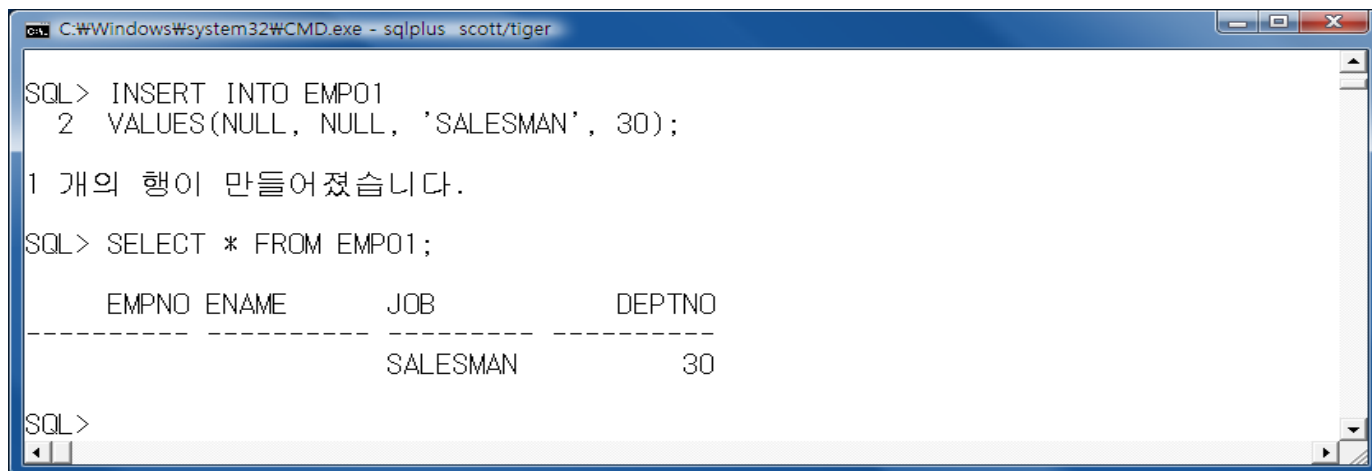


실습

2. EMP01 테이블에 데이터를 추가

```
INSERT INTO EMP01  
VALUES(NULL, NULL, 'SALESMAN', 30);
```

```
SELECT * FROM EMP01;
```



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 EMP01  
2  VALUES(NULL, NULL, 'SALESMAN', 30);  
  
1 개의 행이 만들어졌습니다.  
  
SQL> SELECT * FROM EMP01;
```

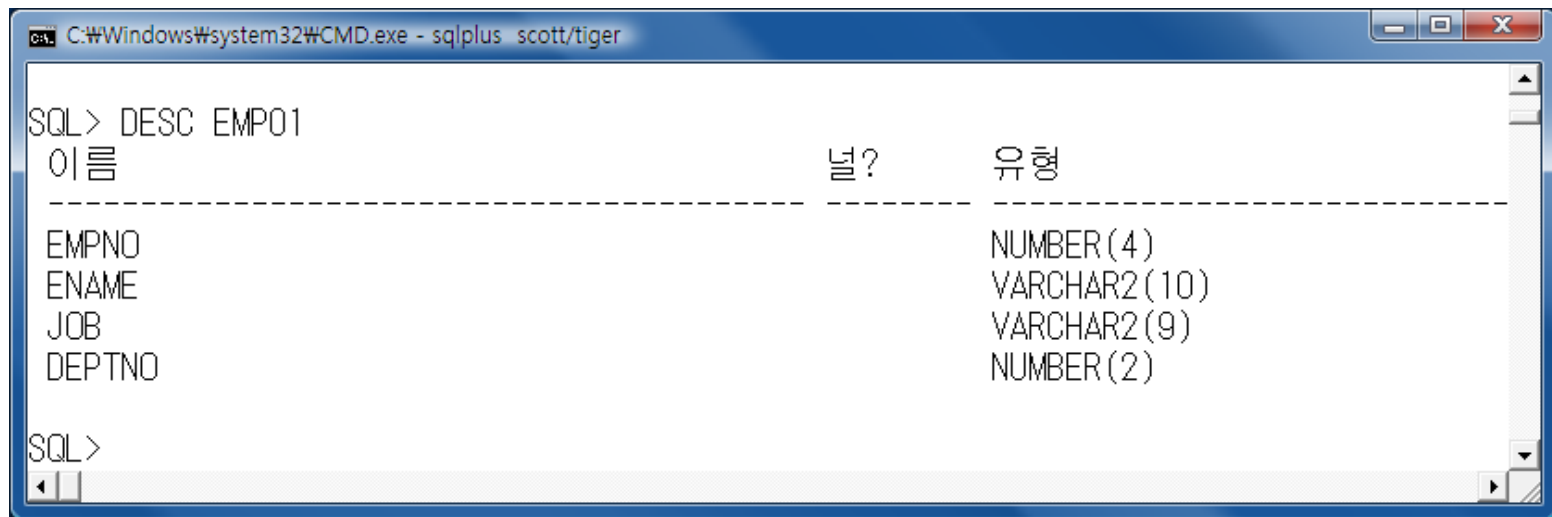
EMPNO	ENAME	JOB	DEPTNO
		SALESMAN	30

The output of the SELECT statement is displayed as a table with four columns: EMPNO, ENAME, JOB, and DEPTNO. The table shows one row with the values (, SALESMAN, 30). The prompt "SQL>" is visible at the bottom left of the window.

실습

3. EMP01 테이블에 사원번호와 사원명에 데이터를 저장하지 않더라도 해당 로우가 테이블에 추가되는데 테이블을 생성하면서 아무런 제약조건도 주지 않았기 때문

DESC EMP01



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\CMD.exe - sqlplus scott/tiger". The command "SQL> DESC EMP01" has been entered. The output displays the structure of the EMP01 table, listing columns and their data types. The columns are EMPNO, ENAME, JOB, and DEPTNO. The data types are NUMBER(4), VARCHAR2(10), VARCHAR2(9), and NUMBER(2) respectively. The output is formatted with a header row and a dashed line separator.

이름	널?	유형
EMPNO		NUMBER(4)
ENAME		VARCHAR2(10)
JOB		VARCHAR2(9)
DEPTNO		NUMBER(2)

3.1 컬럼 레벨 제약조건 지정

- ❖ NOT NULL 제약조건을 지정하지 않으면 위 예에서처럼 NULL 값을 저장 가능
- ❖ 특정 컬럼에 NULL 값이 저장되지 못하도록 하려면 NOT NULL 제약조건을 설정해야 함
- ❖ 제약조건을 설정하는 방법은 컬럼 레벨과 테이블 레벨 두 가지 방식이 있는데 NOT NULL 제약조건은 컬럼 레벨로만 정의 가능

column_name data_type constraint_type



실습하기

- ❖ 사원 테이블(EMP02)을 사원번호, 사원명, 직급, 부서번호 4개의 칼럼으로 구성하되 이번에는 사원번호와 사원명에 NOT NULL 조건을 지정
- 1. 사원 테이블과 유사한 구조의 사원번호, 사원명, 직급, 부서번호 4개의 칼럼으로 구성된 EMP02 테이블을 생성하되 EMPNO와 EMPNAME 컬럼에 NOT NULL 제약조건 설정

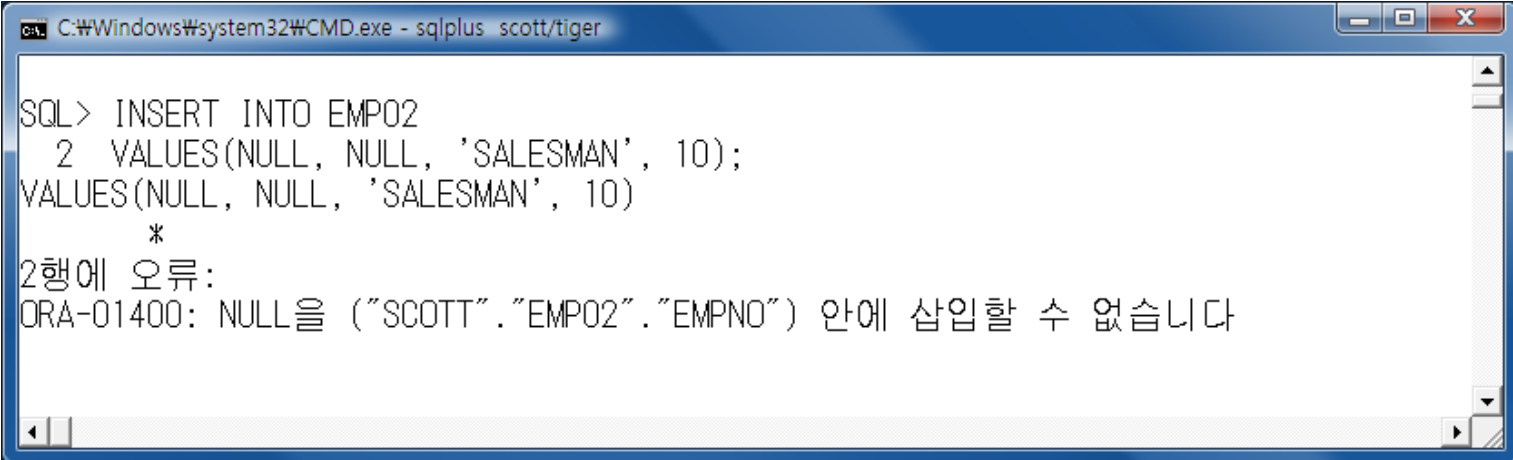
```
CREATE TABLE EMP02(  
    EMPNO NUMBER(4) NOT NULL,  
    ENAME VARCHAR2(10) NOT NULL,  
    JOB VARCHAR2(9),  
    DEPTNO NUMBER(2));
```



실습

2. EMP02 테이블에 데이터를 추가

```
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.

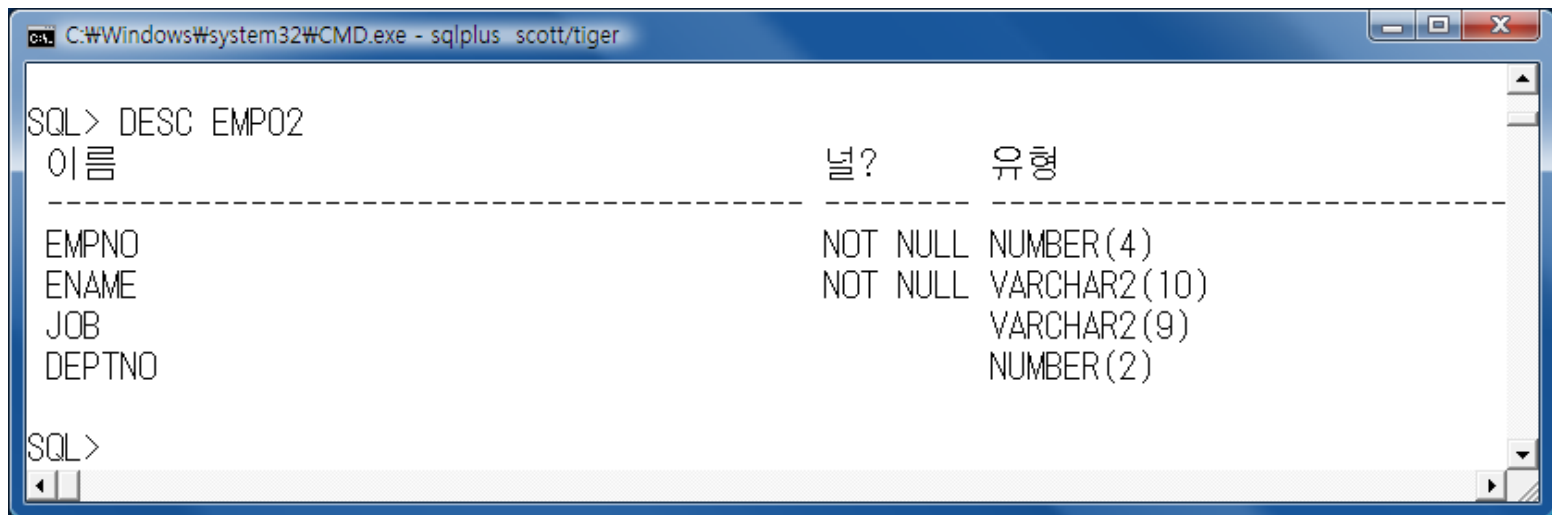
```
C:\Windows\system32\CMD.exe - sqlplus scott/tiger  
SQL> INSERT INTO EMP02  
2 VALUES(NULL, NULL, 'SALESMAN', 10);  
VALUES(NULL, NULL, 'SALESMAN', 10)  
*  
2행에 오류:  
ORA-01400: NULL을 ("SCOTT"."EMP02"."EMPNO") 안에 삽입할 수 없습니다
```

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

실습

3. DESC 명령어로 NOT NULL 제약조건이 설정되어 있음을 확인

DESC EMP02;



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

SQL> DESC EMP02
이름                널?       유형
-----
EMPNO              NOT NULL  NUMBER(4)
ENAME              NOT NULL  VARCHAR2(10)
JOB                VARCHAR2(9)
DEPTNO             NUMBER(2)

SQL>
```

4. UNIQUE

- ❖ UNIQUE 제약조건이란 특정 칼럼에 대해 자료가 중복되지 않게 하는 것
- ❖ 지정된 칼럼에는 유일한 값이 저장



실습

❖ 사원 테이블의 사원번호를 유일한 값으로 지정

1. 사원 테이블과 유사한 구조의 사원번호, 사원명, 직급, 부서번호 4개의 칼럼으로 구성된 EMP03 테이블을 생성하되 사원번호를 유일한 값으로 지정

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



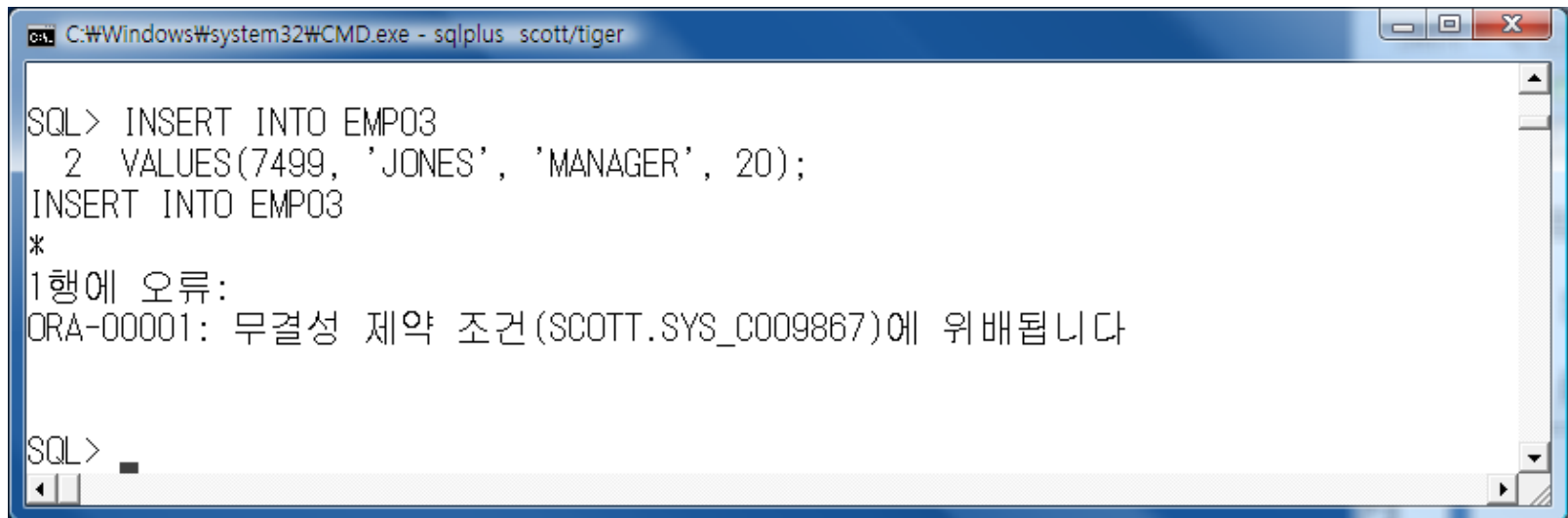
실습

2. EMP03 테이블에 데이터를 추가

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

3. 동일한 사원번호를 추가

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



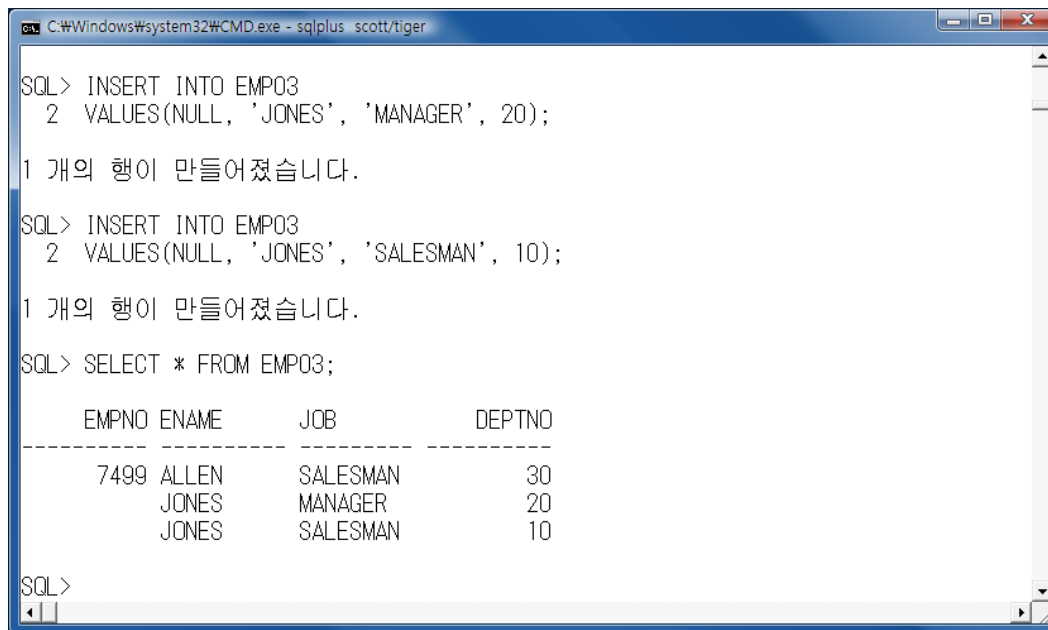
```
C:\Windows\system32\CMD.exe - sqlplus scott/tiger  
  
SQL> INSERT INTO EMP03  
2  VALUES(7499, 'JONES', 'MANAGER', 20);  
INSERT INTO EMP03  
*  
1행에 오류:  
ORA-00001: unique constraint (SCOTT.SYS_C009867)에 위배됩니다  
  
SQL>
```

실습하기

4. NULL 값은 중복되어 저장할 수 있는데 UNIQUE는 값(VALUE)이 유일함을 의미하는 것으로 NULL은 값(VALUE)에서 제외되므로 유일한 조건인지를 체크하는 값에서 제외

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

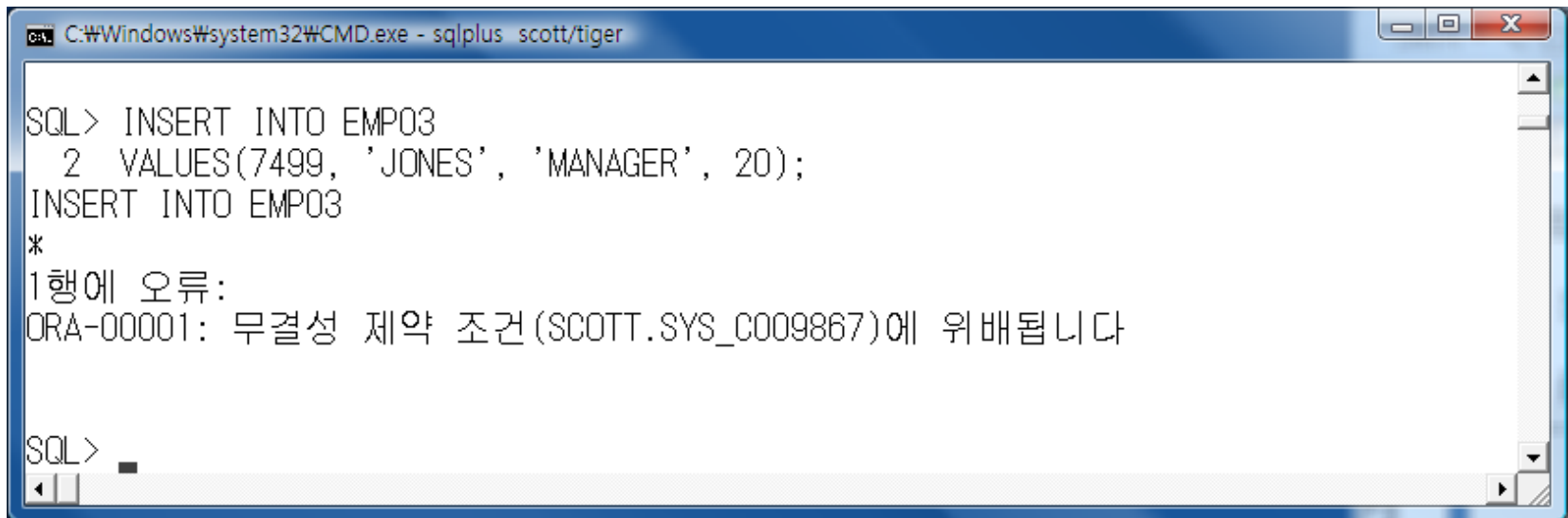
```
INSERT INTO EMP03  
VALUES(NULL, 'JONES', 'SALESMAN', 10);
```



```
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>
```

5. 제약조건 이름 설정

- ❖ 제약조건 이름을 지정하지 않고 제약조건만을 명시하면 오라클 서버가 자동으로 제약조건 이름을 설정
- ❖ 오라클이 부여하는 제약조건 이름은 SYS_ 다음에 숫자를 나열



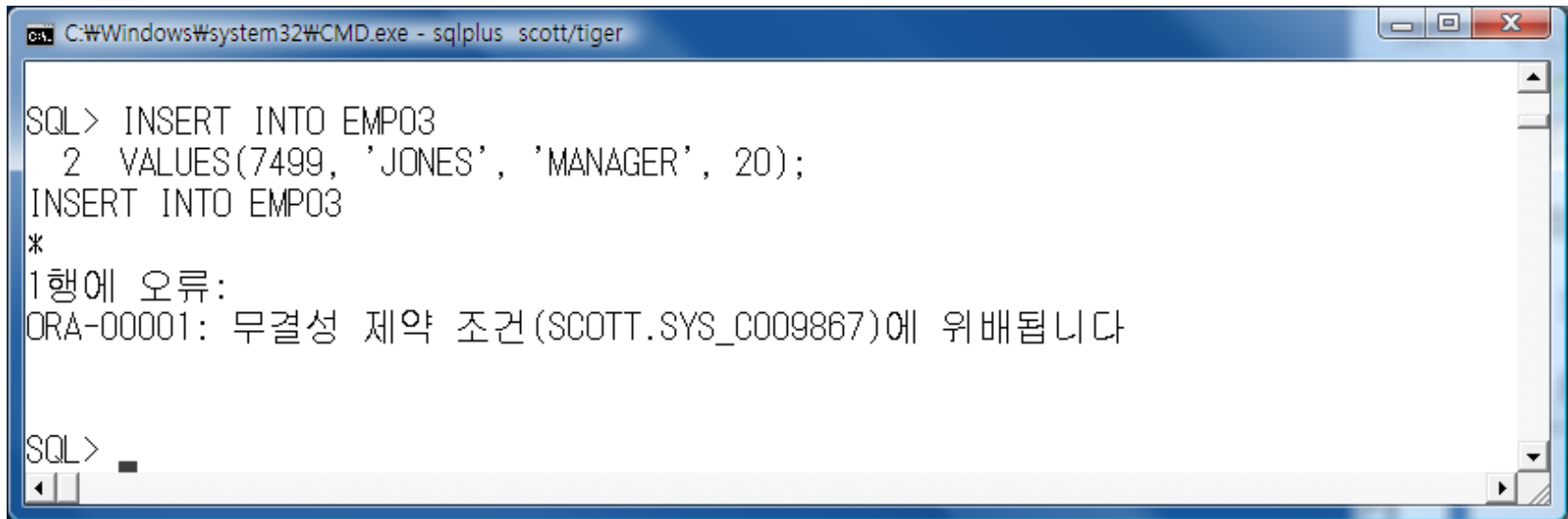
A screenshot of a Windows command prompt window titled "C:\Windows\system32\CMD.exe - sqlplus scott/tiger". The window shows the following text:

```
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>
```

The error message "ORA-00001: 무결성 제약 조건 (SCOTT.SYS_C009867)에 위배됩니다" indicates a unique constraint violation. The background of the slide features a blurred image of a fountain pen and a document.

5. 제약조건 이름 설정

- ❖ 제약조건에 위배하면 오류 메시지에 제약조건 이름만 출력되는데 오라클이 부여한 제약조건 이름으로는 어떤 제약조건을 위반했는지 알 수 없기에 USER_CONSTRAINTS Data Dictionary를 검색해야만 어떤 제약조건인지 확인 할 수 있음
- ❖ 만일 사용자가 의미 있게 제약조건 이름을 명시한다면 제약조건명만으로도 어떤 제약조건을 위배했는지 알 수 있음



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

SQL>
```

The error message "ORA-00001: 무결성 제약 조건 (SCOTT.SYS_C009867)에 위배됩니다" indicates a unique constraint violation. The background of the slide features a faint image of a fountain pen and some documents.

5. 제약조건 이름 설정

- ❖ 제약조건 이름을 지정하는 방법

column_name data_type CONSTRAINT constraint_name constraint_type

- ❖ 사용자 제약조건 이름을 설정하기 위해서는 CONSTRAINT라는 키워드와 함께 제약조건 이름을 기술
- ❖ 제약조건 이름(constraining_name)은 다음과 같은 명명 규칙을 준수해서 작성하는 것을 권장
[테이블명]_[칼럼명]_[제약조건 유형]



실습

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

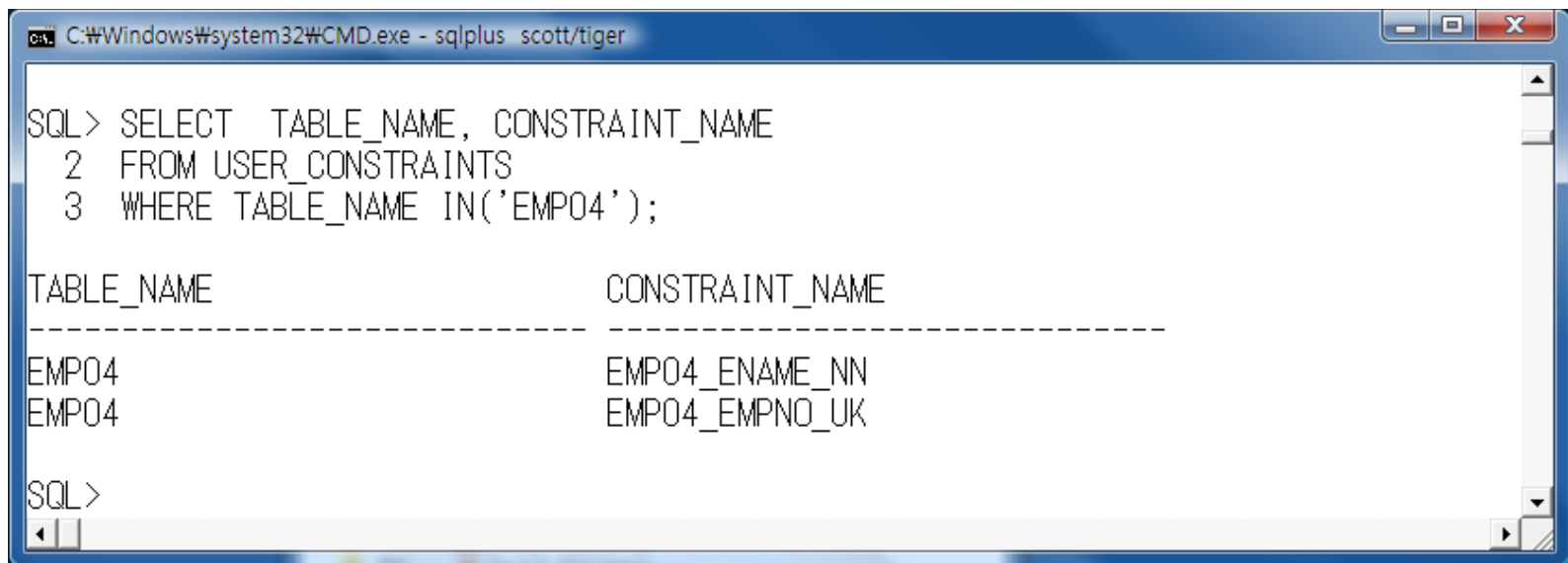
```
CREATE TABLE EMP04(  
    EMPNO NUMBER(4) CONSTRAINT EMP04_EMPNO_UK UNIQUE,  
    ENAME VARCHAR2(10) CONSTRAINT EMP04_ENAME_NN NOT NULL,  
    JOB VARCHAR2(9),  
    DEPTNO NUMBER(2));
```



실습

2. 생성된 제약조건 이름을 확인하기 위해서 USER_CONSTRAINTS 데이터 딕셔너리 뷰를 검색

```
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". Inside the window, the following SQL query is entered and executed:

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

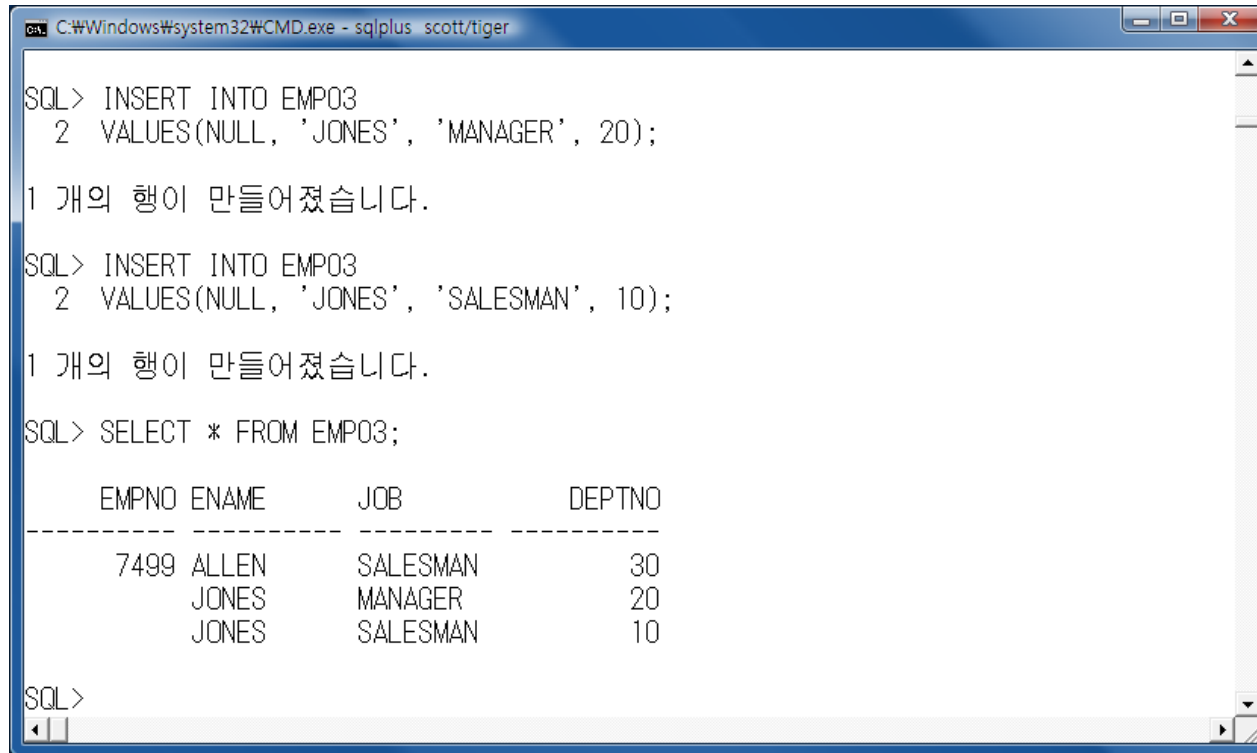
The output of the query is displayed in a table format:

TABLE_NAME	CONSTRAINT_NAME
EMP04	EMP04_ENAME_NN
EMP04	EMP04_EMPNO_UK

The prompt "SQL>" is visible at the bottom of the window.

6. PRIMARY KEY

- ❖ 유일키 제약조건을 지정한 칼럼은 중복된 데이터를 저장하지는 못하지만 NULL 값을 저장하는 것은 허용



```
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>
```

- ❖ 동명이인이 입사를 했다면 이를 구분할 수 있는 유일한 키가 있어야 하는데 사원번호에 NULL 값이 저장되는 바람에 이들을 구분할 수 없음

6. PRIMARY KEY

- ❖ 테이블 내의 해당 행을 다른 행과 구분할 수 있도록 하는 칼럼이 존재하는 것이 좋음
- ❖ 식별 기능을 갖는 칼럼은 유일하면서도 NULL 값을 허용하지 않아야 함
- ❖ UNIQUE 제약조건과 NOT NULL 제약조건을 모두 갖고 있어야 하는데 이러한 두 가지 제약조건을 모두 갖는 것이 기본 키(PRIMARY KEY) 제약조건



실습

❖ 사원 테이블의 사원번호를 기본 키로 지정

1. 사원 테이블과 유사한 구조의 사원번호, 사원명, 직급, 부서번호 4개의 칼럼으로 구성된 테이블을 생성하되 기본 키 제약조건을 설정

```
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)  
);
```



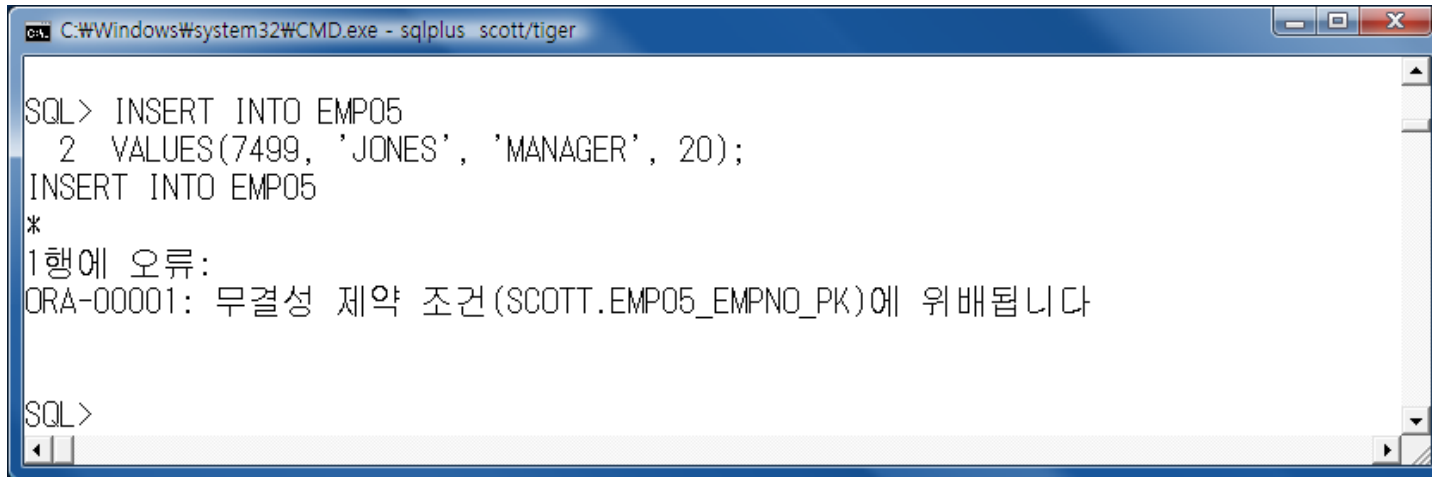
실습

2. 데이터를 추가

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

3. 사원 번호에 동일한 값을 소유한 데이터를 추가

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



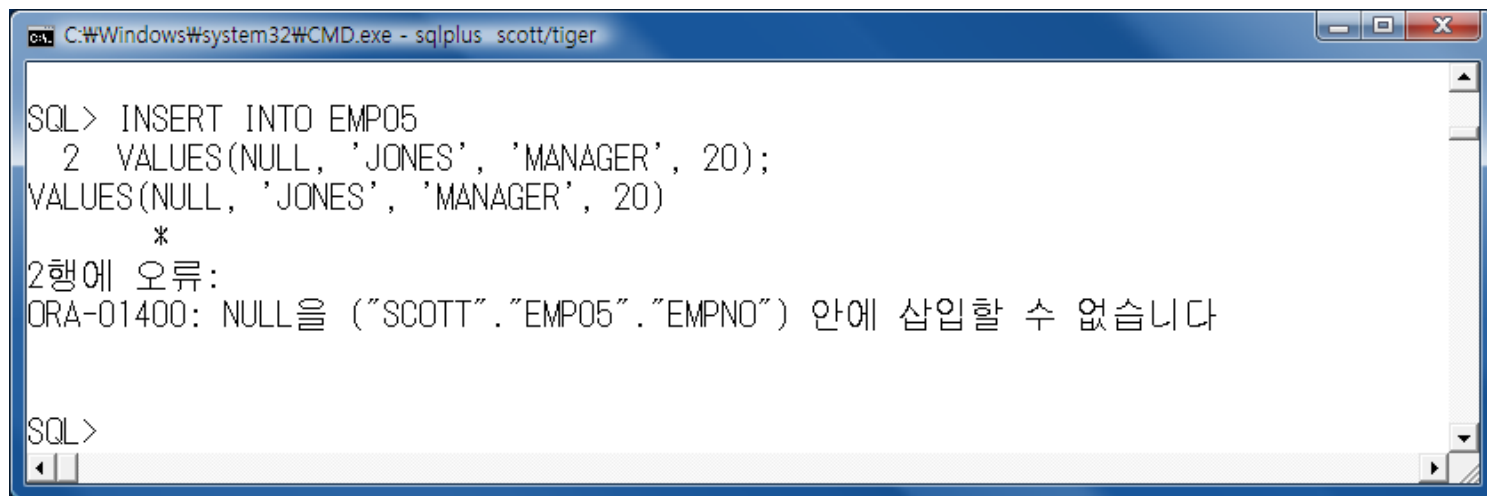
```
C:\Windows\system32\CMD.exe - sqlplus scott/tiger  
  
SQL> INSERT INTO EMP05  
2 VALUES(7499, 'JONES', 'MANAGER', 20);  
INSERT INTO EMP05  
*  
1행에 오류:  
ORA-00001: 무결성 제약 조건 (SCOTT.EMP05_EMPNO_PK)에 위배됩니다  
  
SQL>
```

실습

4. 기본 키로 지정된 사원번호에 NULL 값을 저장

```
INSERT INTO EMP05
```

```
VALUES(NULL, 'JONES', 'MANAGER', 20);
```

A screenshot of a Windows command prompt window titled "C:\Windows\system32\CMD.exe - sqlplus scott/tiger". The window shows an SQL command being entered: "SQL> INSERT INTO EMP05" followed by a new line "2 VALUES(NULL, 'JONES', 'MANAGER', 20);". The command is then executed, resulting in an error message: "ORA-01400: NULL을 ('SCOTT'. 'EMP05'. 'EMPNO') 안에 삽입할 수 없습니다". The prompt "SQL>" is visible at the bottom of the window.

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

SQL> INSERT INTO EMP05
2  VALUES(NULL, 'JONES', 'MANAGER', 20);
VALUES(NULL, 'JONES', 'MANAGER', 20)
      *
2행에 오류:
ORA-01400: NULL을 ('SCOTT'. 'EMP05'. 'EMPNO') 안에 삽입할 수 없습니다

SQL>
```

7. FOREIGN KEY

- ❖ 부서 테이블에는 부서에 대한 정보를 구분하기 위해서 유일하고 NULL이 아닌 값만 저장하도록 부서 번호 컬럼(DEPTNO)를 기본 키로 설정
- ❖ 부서 테이블을 살펴보면 부서 번호가 10, 20, 30, 40인 부서만 존재하는데 부서 테이블의 부서 번호 컬럼(DEPTNO)과 동일한 이름의 컬럼이 사원(EMP) 테이블에도 존재
- ❖ 사원 테이블에 존재하는 부서 번호는 부서 테이블에 존재하는 부서 번호인 10, 20, 30 으로만 저장
- ❖ 해당 회사에 부서가 4개 존재한다면 그 회사에 다니는 직원들도 그 4개의 부서 중에 한 곳에 소속되어야 하기 때문
- ❖ 부서 테이블에 존재하지 않는 부서 번호가 특정 사원의 부서로 지정되어 있다면 이치에 맞지 않음
- ❖ 조인이나 서브 쿼리에서 보듯이 사원 테이블에 없는 상세 정보는 부서 테이블에서 찾아오는데 사원 테이블에 저장된 부서번호가 부서 테이블에 없다면 참조할 때 무결해야 한다는 조건(참조 무결성)에 위배되는 것
- ❖ 사원 테이블에 부서번호를 입력 할 때 부서 테이블에 존재하는 부서번호만 입력하도록 하면 참조 무결성이 지켜지는 것
- ❖ 이를 위해서는 사원 테이블의 부서번호 컬럼에 외래 키 제약조건을 명시해야 함
- ❖ 외래 키 제약조건은 사원 테이블의 부서 번호는 반드시 부서 테이블에 존재하는 부서 번호만 입력하도록 함으로서 사원 테이블이 부서 테이블을 부서 번호로 참조 가능하도록 하는 것

7. FOREIGN KEY

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



- ❖ ERD를 보고 데이터베이스를 구현할 때에는 부서나 사원과 같은 개체는 테이블로 정의하고 소속이란 관계는 참조의 무결성을 위한 특정 컬럼에 외래 키 제약조건으로 정의

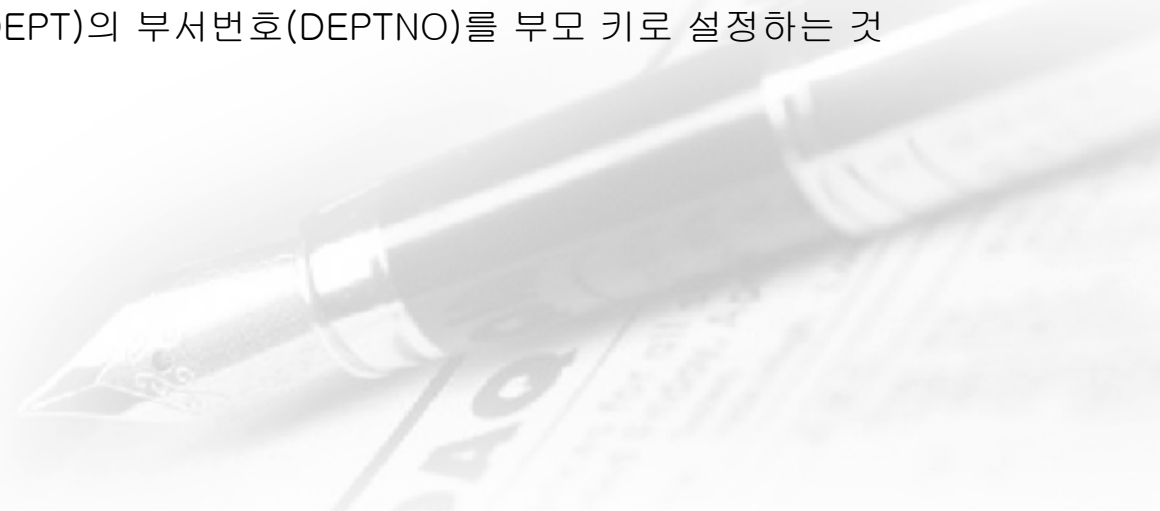


7. FOREIGN KEY

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

사원은 회사 내에 존재하는 부서에 소속되어 있어야 한다

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



7. FOREIGN KEY

- ❖ 부모 키가 되기 위한 칼럼은 반드시 부모 테이블의 기본 키(PRIMARY KEY)나 유일키(UNIQUE)로 설정되어 있어야 함

부모테이블

Primary Key		

자식테이블

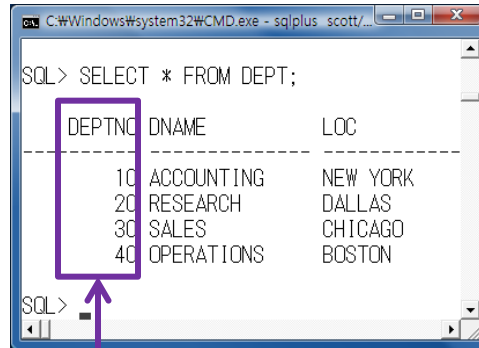
			Foreign Key



- ❖ 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 상태

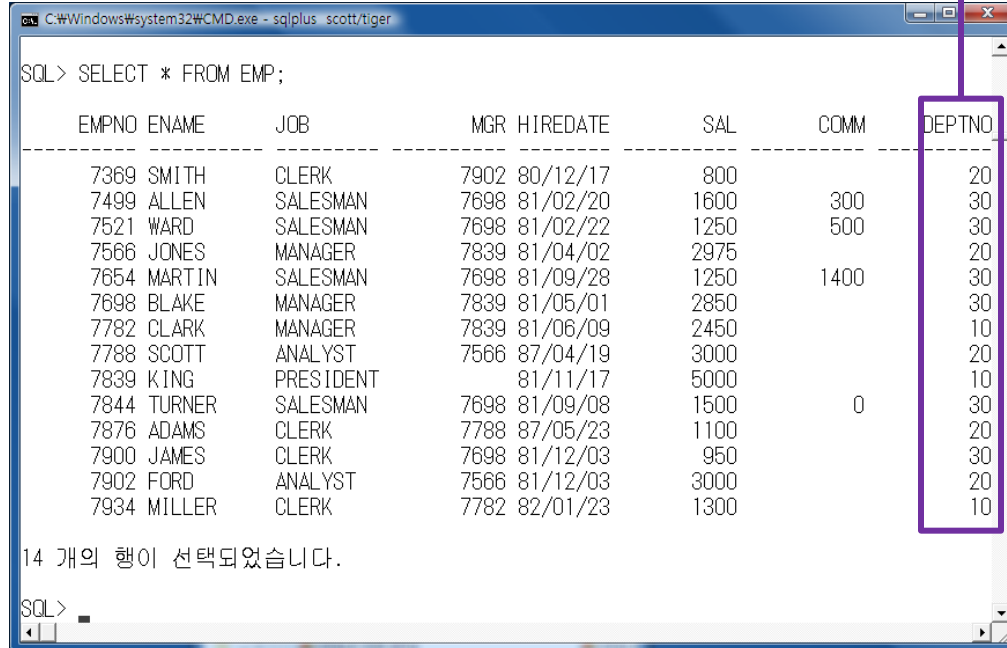
7. FOREIGN KEY

부서(dept) 테이블의 기본 키
인 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> 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.

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

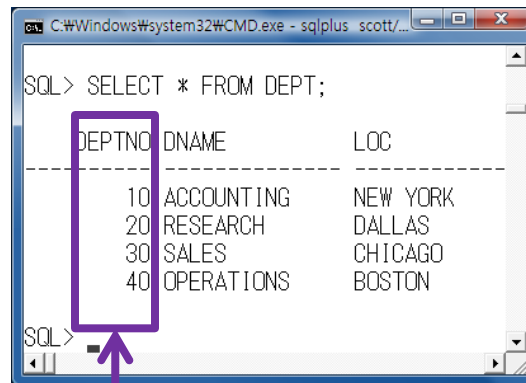
7. FOREIGN KEY

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



7. FOREIGN KEY

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



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

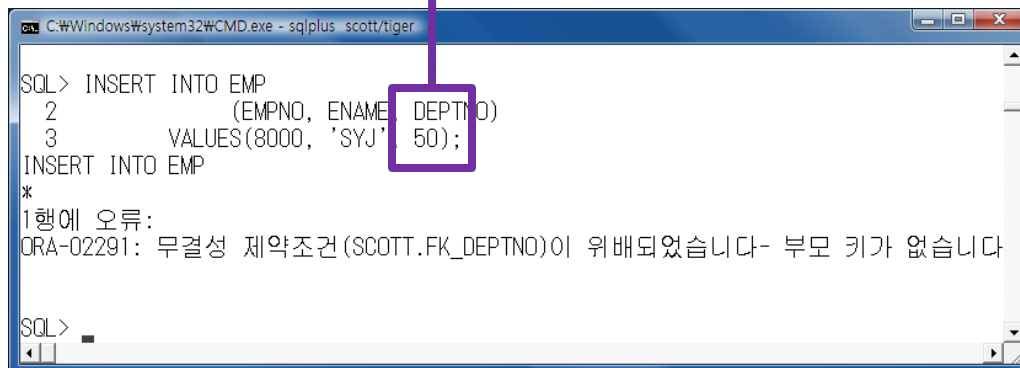
DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

```
SQL>
```

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

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

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



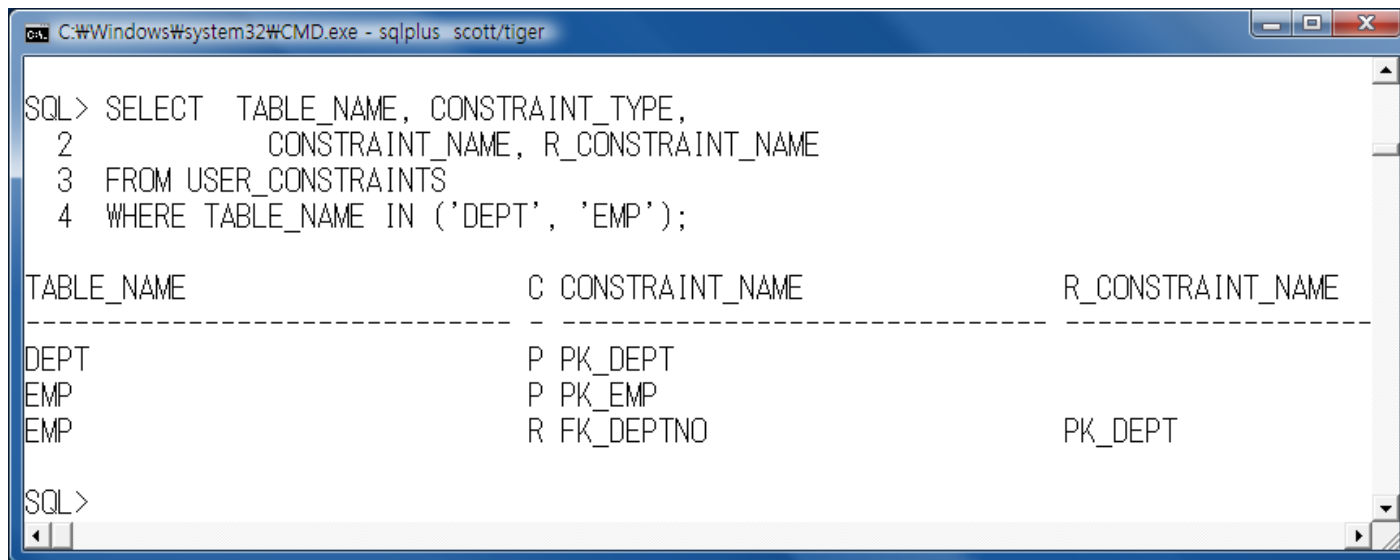
```
SQL> INSERT INTO EMP
2      (EMPNO, ENAME, DEPTNO)
3      VALUES(8000, 'SYJ', 50);
INSERT INTO EMP
*
1행에 오류:
ORA-02291: 무결성 제약조건 (SCOTT.FK_DEPTNO)이 위배되었습니다- 부모 키가 없습니다

SQL>
```

7. FOREIGN KEY

- ❖ EMP 테이블과 DEPT 테이블의 제약조건을 확인

```
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>
```

7. FOREIGN KEY

- ❖ R_CONSTRAINT_NAME 컬럼은 FOREIGN KEY인 경우 어떤 PRIMARY KEY를 참조했는지에 대한 정보를 저장
- ❖ EMP 테이블의 제약조건 FK_DEPTNO의 R_CONSTRAINT_NAME 컬럼값이 PK_DEPT로 설정되어 있음
- ❖ EMP 테이블의 FK_DEPTNO는 외래 키 제약조건으로 PK_DEPT 제약조건을 참조하고 있다는 내용으로 PK_DEPT 제약조건은 DEPT 테이블의 기본 키 제약조건이므로 EMP 테이블은 DEPT 테이블을 참조하고 있는 것



실습

❖ 외래키 제약조건 설정

1. 사원 테이블과 유사한 구조의 사원번호, 사원명, 직급, 부서번호 4개의 칼럼으로 구성된 테이블을 생성하되 기본 키 제약조건은 물론 외래키 제약조건도 설정

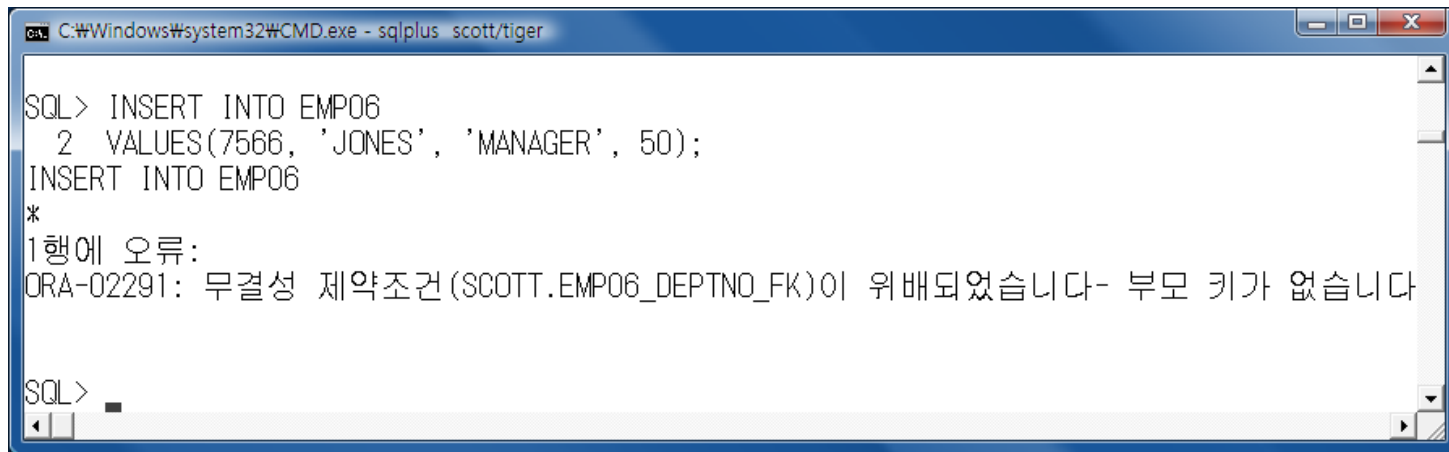
```
CREATE TABLE EMP06(  
    EMPNO NUMBER(4) CONSTRAINT EMP06_EMPNO_PK PRIMARY KEY ,  
    ENAME VARCHAR2(10) CONSTRAINT EMP06_ENAME_NN NOT NULL,  
    JOB VARCHAR2(9),  
    DEPTNO NUMBER(2) CONSTRAINT EMP06_DEPTNO_FK REFERENCES DEPT(DEPTNO)  
);
```



실습

2. EMP06 테이블에 부서 테이블에 존재하지 않는 부서번호를 갖는 사원 정보를 추가

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

A screenshot of a Windows command prompt window titled "C:\Windows\system32\CMD.exe - sqlplus scott/tiger". The window shows the execution of an SQL INSERT statement. The text inside the window is as follows:

```
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> 
```

The error message is in Korean, indicating a foreign key constraint violation. The background of the slide features a faint image of a fountain pen and some papers.

7. FOREIGN KEY

- ❖ ON DELETE CASCADE를 사용하면 상위 테이블에서 행을 삭제하면 하위 테이블에서 데이터가 자동으로 삭제되도록 할 수 있음

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

- ❖ ON DELETE SET NULL은 삭제될 때 하위 테이블의 데이터를 null로 변경
- ❖ ON UPDATE CASCADE도 있음



실습

❖ 외래키 제약조건 설정

1. 사원 테이블과 유사한 구조의 사원번호, 사원명, 직급, 부서번호 4개의 칼럼으로 구성된 테이블을 생성하되 기본 키 제약조건은 물론 외래키 제약조건도 설정

```
CREATE TABLE EMP07(  
    EMPNO NUMBER(4) CONSTRAINT EMP07_EMPNO_PK PRIMARY KEY ,  
    ENAME VARCHAR2(10) CONSTRAINT EMP07_ENAME_NN NOT NULL,  
    JOB VARCHAR2(9),  
    DEPTNO NUMBER(2)CONSTRAINT EMP07_DEPTNO_FK REFERENCES  
        DEPT(DEPTNO)ON DELETE CASCADE  
);
```



8. CHECK

- ❖ CHECK 제약조건은 입력되는 값을 체크하여 설정된 값 이외의 값이 들어오면 오류 메시지와 함께 명령이 수행되지 못하게 하는 것
- ❖ 조건으로 데이터의 값의 범위나 특정 패턴의 숫자나 문자 값을 설정할 수 있음
- ❖ 사원 테이블에 급여 컬럼을 생성하되 급여 컬럼 값은 500에서 5000사이의 값만 저장할 수 있도록 하거나 성별을 저장하는 컬럼으로 GENDER를 정의하고 이 컬럼에는 남자는 M, 여자는 F 둘 중의 하나만 저장할 수 있도록 제약을 주려면 CHECK 제약조건을 지정



실습

- ❖ 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 지정

1. 사원번호, 사원명, 직급, 부서번호, 직급, 성별 6개의 칼럼으로 구성된 테이블을 생성하되 기본 키 제약조건, 외래키 제약조건은 물로 CHECK 제약조건도 설정

```
CREATE TABLE EMP08(  
    EMPNO NUMBER(4) CONSTRAINT EMP08_EMPNO_PK PRIMARY KEY ,  
    ENAME VARCHAR2(10) CONSTRAINT EMP08_ENAME_NN NOT NULL,  
    SAL NUMBER(7, 2)CONSTRAINT EMP08_SAL_CK CHECK(SAL BETWEEN 500 AND 5000),  
    GENDER VARCHAR2(1) CONSTRAINT EMP08_GENDER_CK CHECK (GENDER IN('M', 'F'))  
);
```



9. DEFAULT 제약조건

- ❖ 디폴트는 아무런 값을 입력 하지 않았을 때 디폴트 제약의 값이 입력됨
- ❖ 자료형 뒤에 DEFAULT 기본값 의 형식으로 설정하면 됨



실습

- ❖ 만약 지역명(LOC)라는 컬럼에 아무런 값도 입력 안했을 때 디폴트의 값인 'SEOUL'이 들어가도록 하고 싶을 경우 디폴트 제약조건을 지정

1. 다음과 같이 부서 테이블을 생성

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

2. 지역명(LOC)라는 컬럼에 아무런 값도 입력하지 않았을 때 디폴트의 값인 'SEOUL'이 들어감을 확인

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

```
SELECT *  
FROM DEPT01;
```



10. 테이블 레벨 제약조건

- ❖ 지금까지 제약조건을 지정하는 방식을 컬럼 레벨의 제약조건 지정
- ❖ 컬럼 레벨 제약조건
 - ✓ CREATE TABLE로 테이블을 생성하면서 컬럼을 정의하게 되는데 하나의 컬럼 정의가 다 마무리되기 전에 컬럼 명 다음에 타입을 지정하고 그 뒤에 연이어서 제약조건을 지정하는 방식
- ❖ 테이블 레벨의 제약조건
 - ✓ 컬럼을 모두 정의하고 나서 테이블 정의를 마무리 짓기 전에 따로 생성된 컬럼들에 대한 제약조건을 한꺼번에 지정하는 것
- ❖ 일반적으로 컬럼 레벨 방식으로 제약조건을 지정하는 것이 간편한데 테이블 레벨의 지정 방식을 사용하는 데에는 2가지 이유가 있음
 - ✓ 복합키로 기본키를 지정할 경우
 - 지금까지는 한 개의 컬럼으로 기본키를 지정했지만 경우에 따라서는 2개 이상의 컬럼이 하나의 기본키를 구성하는 경우가 있는데 이를 복합키라고 하는데 복합키 형태로 제약조건을 지정할 경우에는 컬럼 레벨 형식으로는 불가능하고 반드시 테이블 레벨 방식을 사용
 - ✓ ALTER TABLE로 제약조건을 추가할 때
 - 테이블의 정의가 완료되어서 이미 테이블의 구조가 결정된 후에 나중에 테이블에 제약조건을 추가하고 할 때에는 테이블 레벨 방식으로 제약조건을 지정

10. 테이블 레벨 제약조건

- ❖ 테이블 레벨 정의 방식의 기본 형식

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

- ❖ 테이블 레벨에서 칼럼의 제약조건을 정의할 때 NOT NULL 조건은 테이블 레벨 정의 방법으로 제약조건을 지정할 수 없음



실습

❖ 컬럼 레벨로 제약조건을 지정하는 것과 테이블 레벨로 제약조건을 지정

1. 컬럼 레벨로 제약조건 지정

```
CREATE TABLE EMP11(  
    EMPNO NUMBER(4) PRIMARY KEY,  
    ENAME VARCHAR2(10) NOT NULL,  
    JOB VARCHAR2(9) UNIQUE,  
    DEPTNO NUMBER(4) REFERENCES DEPT(DEPTNO));
```

2. 테이블 레벨로 제약조건 지정

```
CREATE TABLE EMP12(  
    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)  
);
```



실습

3. 제약조건 이름을 지정하여 테이블 레벨 방식으로 제약조건을 지정

```
CREATE TABLE EMP13(  
    EMPNO NUMBER(4) CONSTRAINT EMP13_ENAME_NN NOT NULL,  
    ENAME VARCHAR2(10),  
    JOB VARCHAR2(9),  
    DEPTNO NUMBER(4),  
    CONSTRAINT EMP13_EMPNO_PK PRIMARY KEY(EMPNO),  
    CONSTRAINT EMP13_JOB_UK UNIQUE(JOB),  
    CONSTRAINT EMP13_DEPTNO_FK FOREIGN KEY(DEPTNO) REFERENCES  
    DEPT(DEPTNO));
```



실습

❖ 복합키를 기본 키로 지정

1. 이름과 핸드폰 번호를 복합하여 이를 기본키로 지정

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



11.1 제약조건 추가

- ❖ 테이블 생성이 끝난 후에 제약조건을 추가하기 위해서는 ALTER TABLE로 추가
- ❖ 새로운 제약조건을 추가하려면 ALTER TABLE 문에 ADD 절을 사용
- ❖ 다음은 제약조건을 추가하기 위한 형식

```
ALTER TABLE table_name  
ADD [CONSTRAINT constraint_name] constraint_type (column_name);
```



실습

❖ 제약조건 없이 테이블을 생성한 후 제약조건을 추가

1. 사원 테이블과 유사한 구조의 사원번호, 사원명, 직급, 부서번호 4개의 칼럼으로 구성된 EMP01 테이블을 제약조건을 하나도 설정하지 않은 채 생성

```
CREATE TABLE EMP21(  
    EMPNO NUMBER(4),  
    ENAME VARCHAR2(10),  
    JOB VARCHAR2(9),  
    DEPTNO NUMBER(4));
```

2. EMP21 테이블에 2가지 제약조건을 설정하는데 첫 번째는 EMPNO 컬럼에 기본키를 설정하고 두 번째에는 DEPTNO 컬럼에 외래키를 설정

```
ALTER TABLE EMP21  
ADD CONSTRAINT EMP21_EMPNO_PK PRIMARY KEY(EMPNO);
```

```
ALTER TABLE EMP21  
ADD CONSTRAINT EMP21_DEPTNO_FK FOREIGN KEY(DEPTNO) REFERENCES  
DEPT(DEPTNO);
```

11.2 MODIFY로 제약조건 추가

- ❖ NOT NULL 제약조건을 이미 존재하는 테이블에 추가하는 경우 이미 존재하는 테이블에 무결성 제약조건을 추가로 생성하기 위해서 ALTER TABLE ... ADD ... 명령문을 사용
- ❖ NOT NULL 제약조건은 ADD 대신 MODIFY 명령문을 사용
- ❖ 이는 'NULL을 허용하는 상태'에서 'NULL을 허용하지 않는 상태'로 변경하는 것이기 때문
- ❖ MODIFY 명령어로 NOT NULL 제약조건을 설정
ALTER TABLE EMP21
MODIFY ENAME CONSTRAINT EMP21_ENAME_NN NOT NULL;



11.3 제약조건 제거하기

- ❖ 제약조건을 제거하기 위해서 DROP CONSTRAINT 다음에 제거하고자 하는 제약조건 이름을 명시
- ❖ 제약조건을 제거하기 위한 형식
ALTER TABLE table_name
DROP [CONSTRAINT constraint_name];
- ❖ 제약조건을 CONSTRAINT 문을 사용하여 지정했을 경우에는 제약조건 이름을 기억하기 쉽지만 CONSTRAINT 문을 사용하지 않았으면 특정 테이블의 특정 컬럼에 명시된 제약조건을 USER_CONSTRAINTS 데이터 디렉터리 뷰에서 찾아봐야 하는 불편함이 있음
- ❖ 제약조건을 지정할 때에는 제약조건이름을 명시적으로 주는 것이 바람직



실습

❖ 사원 테이블에 지정한 제약조건들을 제거

1. 기본 키 제약조건을 제거

```
ALTER TABLE EMP21  
DROP CONSTRAINT EMP21_EMPNO_PK;
```

2. 사원명에 NULL이 저장될 수 있도록 NOT NULL 제약조건을 제거

```
ALTER TABLE EMP21  
DROP CONSTRAINT EMP21_ENAME_NN;
```



12. 비활성화와 CASCADE

- ❖ 제약조건이 설정되면 항상 그 규칙에 따라 데이터 무결성이 보장
- ❖ 특별한 업무를 수행하는 과정에서 이러한 제약조건 때문에 작업이 진행되지 못하는 경우가 생기는데 그렇다고 제약조건을 삭제해 버리면 데이터 무결성을 보장받지 못하게 됨
- ❖ 오라클에서는 제약조건을 비활성화시킴으로서 제약조건을 삭제하지 않고도 제약조건 사용을 잠시 보류할 수 있는 방법을 제공
- ❖ 비활성화 된 제약조건은 원하는 작업을 한 후에는 다시 활성화 상태로 만들어 주어야 함



실습

- ❖ 제약조건을 비활성화를 위한 실습을 위해서 부서 테이블을 만든 후에 부서 테이블을 부모 테이블로 하는 사원 테이블을 생성하는데 부서 테이블의 부서번호가 기본 키로 설정되어 있고 사원 테이블의 부서번호가 부서 테이블의 부서번호를 참조할 수 있도록 외래 키를 설정

1. 부서 테이블에 부서번호를 기본 키로 지정하여 새로운 테이블을 생성

```
CREATE TABLE DEPT31(  
    DEPTNO NUMBER(2) CONSTRAINT DEPT01_DEPTNO_PK PRIMARY KEY,  
    DNAME VARCHAR2(14),  
    LOC VARCHAR2(13)  
);
```

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

```
CREATE TABLE EMP31(  
    EMPNO NUMBER(4) CONSTRAINT EMP31_EMPNO_PK PRIMARY KEY ,  
    ENAME VARCHAR2(10) CONSTRAINT EMP31_ENAME_NN NOT NULL,  
    JOB VARCHAR2(9),  
    DEPTNO NUMBER(4) CONSTRAINT EMP31_DEPTNO_FK REFERENCES  
    DEPT31(DEPTNO)  
);
```

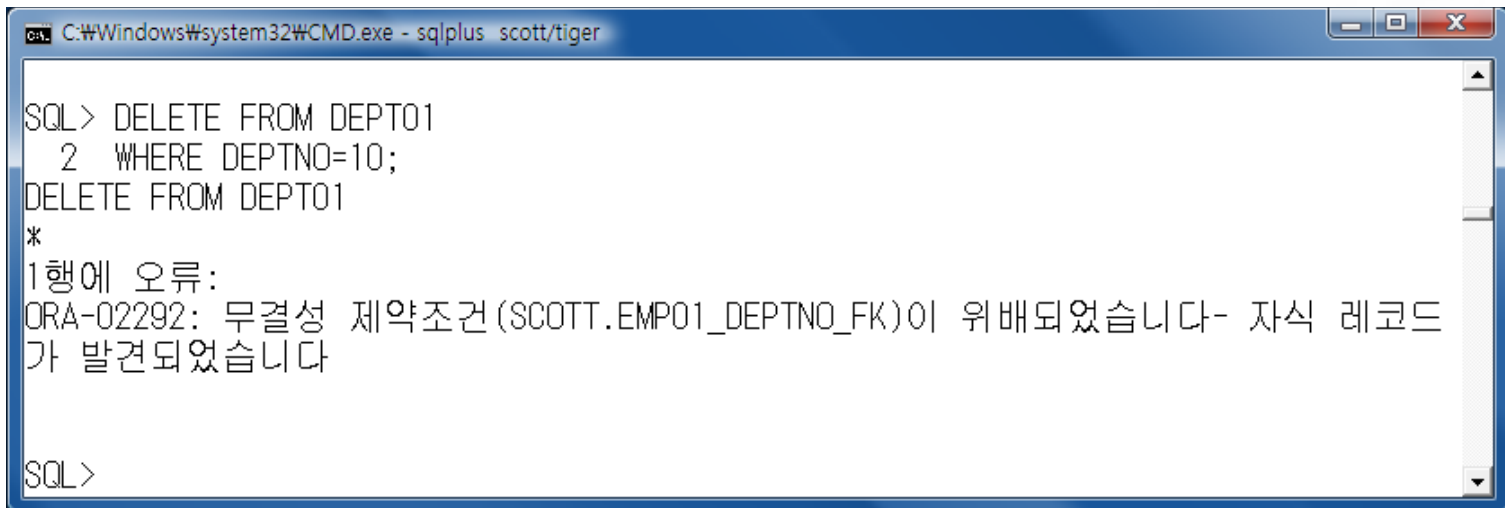
실습

3. 데이터 추가

```
INSERT INTO DEPT31 VALUES(10, 'ACCOUNT', 'SEOUL');  
INSERT INTO DEPT31 VALUES(20, 'SALES', 'MOKPO');
```

```
INSERT INTO EMP31 VALUES(7499, 'ALLEN', 'SALESMAN', 10);  
INSERT INTO EMP31 VALUES(7369, 'SMITH', 'CLERK', 20);
```

4. DEPT01 테이블에서 10번 부서를 ALLEN이란 사람이 참조하고 있는 상태에서 삭제
DELETE FROM DEPT31 WHERE DEPTNO=10;

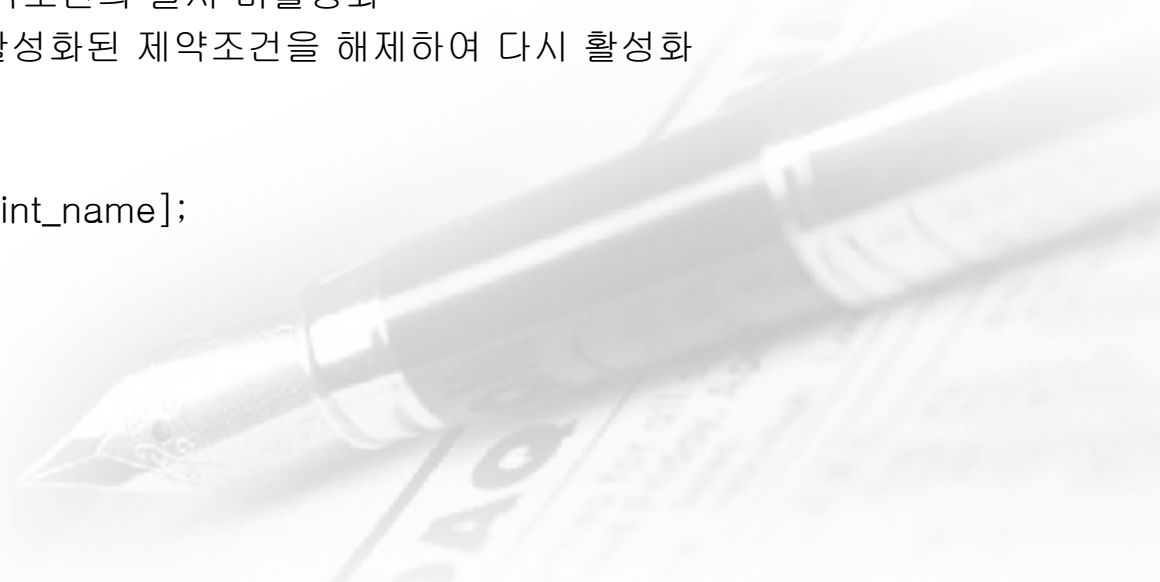


```
C:\Windows\system32\CMD.exe - sqlplus scott/tiger  
  
SQL> DELETE FROM DEPT01  
2 WHERE DEPTNO=10;  
DELETE FROM DEPT01  
*  
1행에 오류:  
ORA-02292: 무결성 제약조건 (SCOTT.EMP01_DEPTNO_FK)이 위배되었습니다- 자식 레코드가 발견되었습니다  
  
SQL>
```

12.1 제약조건의 비활성화

- ❖ 자식 테이블인 사원 테이블(EMP01)은 부모 테이블인 부서 테이블(DEPT01)에 기본 키인 부서번호를 참조하고 있어서 부서 테이블의 10번 부서는 사원 테이블에 근무하는 10번 사원이 존재하기 때문에 삭제할 수 없음
- ❖ 부모 테이블(DEPT01)의 부서번호 10번이 삭제되면 자식 테이블(EMP01)에서 자신이 참조하는 부모를 잃어버리게 되므로 삭제할 수 없는 것
- ❖ 부서번호가 10인 자료가 삭제되도록 하기 위해서는 자식 테이블의 데이터를 먼저 삭제하던지 아니면 외래키를 삭제
- ❖ 테이블에서 제약조건을 삭제하지 않고 일시적으로 적용시키지 않도록 하는 방법으로 제약조건을 비활성화하는 방법이 있음
- ❖ 오라클은 다음 명령어로 작업 상황에 따라 비활화시키거나 다시 활성화시키면서 유연하게 작업할 수 있도록 함
 - ✓ DISABLE CONSTRAINT : 제약조건의 일시 비활성화
 - ✓ ENABLE CONSTRAINT : 비활성화된 제약조건을 해제하여 다시 활성화

```
ALTER TABLE table_name  
DISABLE [CONSTRAINT constraint_name];
```



12.1 제약조건의 비활성화

- ❖ EMP01 테이블에 지정한 제약조건 중에서 외래키 제약조건이 있음
- ❖ 이 제약조건 때문에 DEPT31 테이블에서 10번 부서를 삭제할 수 없음
- ❖ EMP01 테이블의 ALLEN이란 사람이 DEPT31 테이블에서 10번 부서를 참조하고 있는 상태였기 때문
- ❖ EMP01 테이블에 지정한 외래키 제약조건을 비활성화 시키고 나면 EMP31 테이블과 DEPT31 테이블이 아무런 관계도 없는 상태가 되기 때문에 DEPT31 테이블에서 10번 부서를 삭제하는 데 아무런 문제가 없음



실습

- ❖ EMP31 테이블에 지정한 외래키 제약조건을 비활성화한 후에 DEPT01 테이블에서 10번 부서를
- ❖ EMP31 테이블에 지정한 외래키 제약조건을 비활성화

```
ALTER TABLE EMP31  
DISABLE CONSTRAINT EMP31_DEPTNO_FK;
```

- ❖ 제약조건을 상태를 확인하기 위해서 USER_CONSTRAINTS 데이터 디렉터리의 STATUS 칼럼값을 살펴보면 EMP31_DEPTNO_FK 제약조건에 대해서 STATUS 칼럼값이 DISABLED로 지정되어 있음을 확인

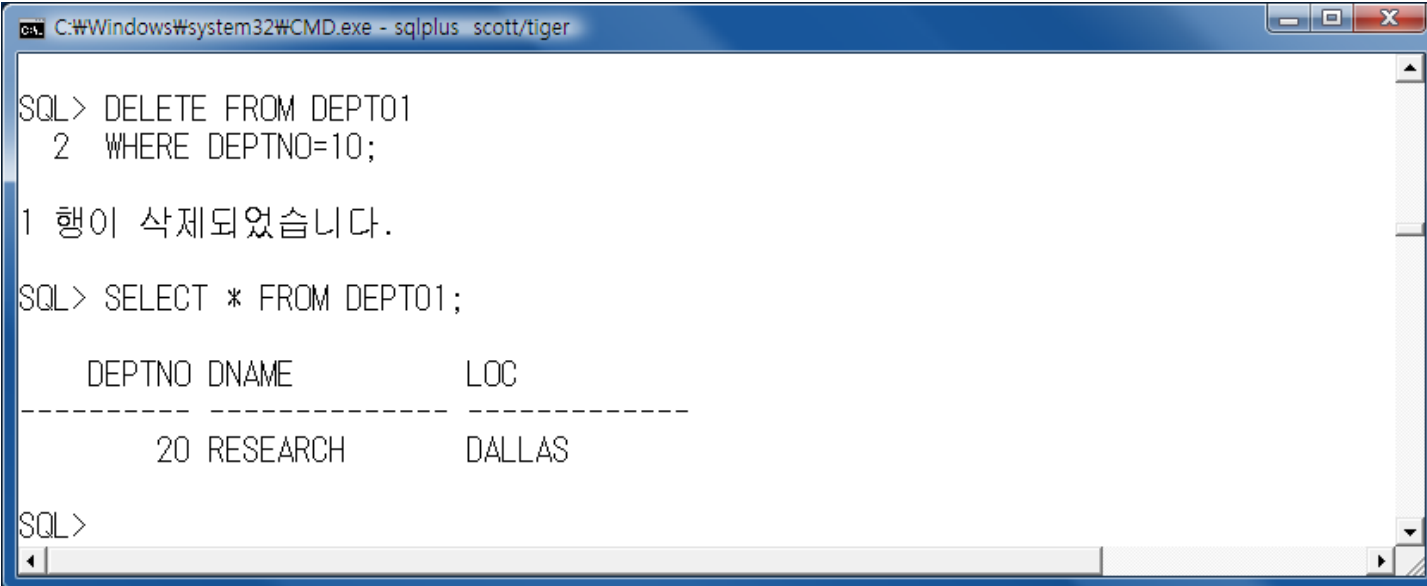
```
SELECT CONSTRAINT_NAME, CONSTRAINT_TYPE,  
TABLE_NAME, R_CONSTRAINT_NAME, STATUS  
FROM USER_CONSTRAINTS  
WHERE TABLE_NAME='EMP31';
```



실습

- ❖ EMP31 테이블에 지정한 외래키 제약조건을 비활성화하였기 때문에 DEPT31 테이블에서 10번 부서를 삭제할 수 있음

```
DELETE FROM DEPT31  
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>
```

12.2 제약조건의 활성화

- ❖ EMP31 테이블에서 비활성화된 제약조건은 다시 활성화
- ❖ DISABLE CONSTRAINT 문에 의해 비활성화된 제약조건을 되살리려면 ENABLE 을 사용
ALTER TABLE EMP31
ENABLE CONSTRAINT EMP31_DEPTNO_FK;
- ❖ 부서 테이블의 10번 부서가 삭제된 상태에서는 외래키 제약조건을 활성화 시킬 수 없음
- ❖ 외래키 제약조건은 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 때문



실습

- ❖ 외래키 제약조건을 활성화시키기 전에 먼저 삭제된 부서 테이블의 10번 부서를 새로 입력

```
INSERT INTO DEPT31  
VALUES(10, 'ACCOUNTING', 'NEW YORK');
```

```
SELECT * FROM DEPT31;
```

- ❖ 10번 부서를 새로 입력해 놓았으므로 이제 외래키 제약조건을 활성화

```
ALTER TABLE EMP31  
ENABLE CONSTRAINT EMP31_DEPTNO_FK;
```



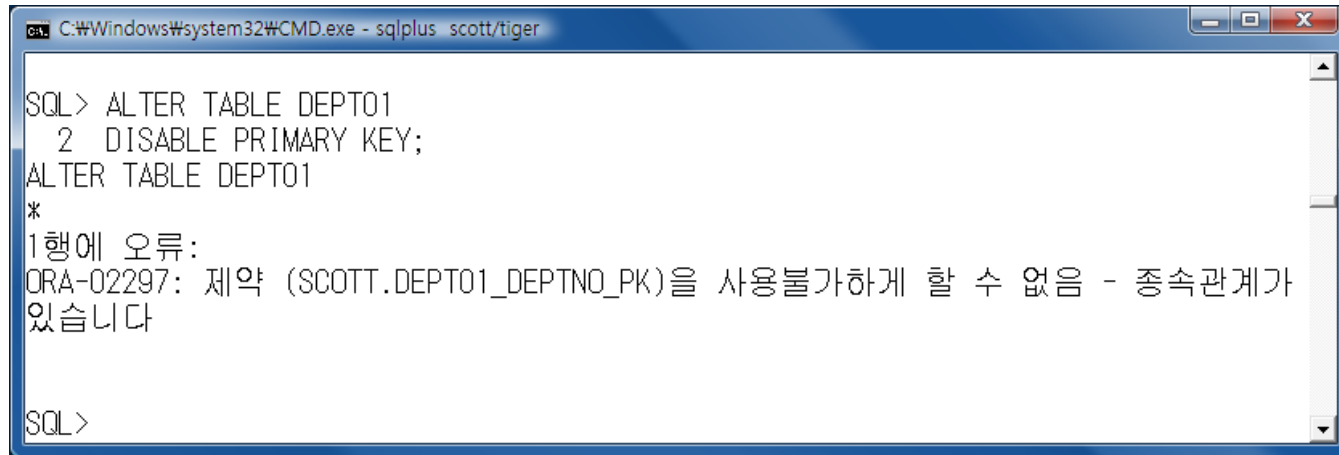
12.3 CASCADE 옵션

- ❖ CASCADE 옵션은 부모 테이블과 자식 테이블 간의 참조 설정이 되어 있을 때 부모 테이블의 제약 조건을 비활성화하면 이를 참조하고 있는 자식 테이블의 제약조건까지 같이 비활성화시켜 주는 옵션
- ❖ 제약조건의 비활성화뿐만 아니라 제약조건이 삭제에도 활용되며, 역시 같은 이치로 부모 테이블의 제약조건을 삭제하면 이를 참조하고 있는 자식 테이블의 제약조건도 같이 삭제



실습

- ❖ 부서 테이블(DEPT31)의 기본 키 제약조건을 비활성화
- ❖ 부서 테이블(DEPT31)의 기본 키 제약조건을 "DISABLE PRIMARY KEY"로 비 활성화하려고 시도
ALTER TABLE DEPT31
DISABLE PRIMARY KEY;



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

```
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 제약조건을 비활성화할 수 없음

12.3 CASCADE 옵션

- ❖ 부모 테이블(부서)의 기본 키에 대한 제약조건을 비활성화하고자 하는 것인데 자식 테이블(사원)에서 이를 외래 키 제약조건으로 지정한 컬럼이라면 비활성화할 수 없음
- ❖ 만일 비활성화될 수 있다고 가정하면 기본 키가 더 이상 아닌 상태 일반 컬럼을 자식 테이블이 외래 키 제약조건으로 지정하고 있는 아이러니 한 상태가 되기 때문
- ❖ 그렇기 때문에 부모 테이블(부서)의 기본 키에 대한 제약조건을 비활성화하려면 자식 테이블(사원)의 외래 키에 대한 제약조건을 비활성화하는 작업을 선행
- ❖ 두 테이블 사이에 아무런 관련이 없어야 만 즉, 부서 테이블이 더 이상 부모 테이블로서의 역할을 하지 않고 있어야만 기본 키 제약조건을 비활성화 시킬 수 있음
 - ✓ 부모 테이블의 기본 키를 참조하는 자식 테이블의 외래 키에 대한 제약조건을 비활성화
 - ✓ 부모 테이블의 기본 키에 대한 제약조건을 비활성화
- ❖ 위 순서대로 제약조건을 여러 번에 걸쳐 비활성화 시키기는 번거로움을 없애주는 것이 CASCADE 옵션인데 기본 키 제약조건을 비활성화하면 이를 참조하는 외래 키 제약조건도 연속적으로 비활성화되기 때문에 한 번만 비활성화 해 주면 됨



실습

- ❖ CASCADE 옵션을 지정하여 기본 키 제약조건을 비활성화하면 이를 참조하는 외래 키 제약조건도 연속적으로 비활성화
 - ✓ 부서 테이블(DEPT01)의 기본 키 제약조건을 CASCADE 옵션을 지정하여 비활성화합시다.
ALTER TABLE DEPT31
DISABLE PRIMARY KEY CASCADE;
 - ✓ 데이터 덱서너리 USER_CONSTRAINTS를 확인
SELECT CONSTRAINT_NAME, CONSTRAINT_TYPE,
TABLE_NAME, R_CONSTRAINT_NAME, STATUS
FROM USER_CONSTRAINTS
WHERE TABLE_NAME IN('DEPT31', 'EMP31');

