



Database 프로그래밍을 위한
오라클 명령어

06

데이터 조작어 (DML)

강 사 : 김 진 성



목 차

1

INSERT

2

다중 테이블에 다중 행 입력하기

3

UPDATE

4

DELETE



01. INSERT 문

- INSERT 문은 테이블에 새로운 데이터를 입력하기 위해 사용하는 데이터 조작용어
- 다음은 INSERT 문의 기본 형식

```
INSERT INTO table_name  
(column_name, ...)  
VALUES(column_value, ...);
```



<실습하기> Insert문 실습을 위한 테이블 생성

1. DEPT01 테이블 생성

```
CREATE TABLE DEPT01(  
  DEPTNO NUMBER(4),  
  DNAME VARCHAR2(30),  
  LOC VARCHAR2(20)  
);
```



빈 테이블 데이터 추가하기

- 새로운 데이터를 추가하기 위해서 사용할 명령어 INSERT INTO ~ VALUES ~는 칼럼 명에 기술된 목록의 수와 VALUES 다음에 나오는 괄호에 기술한 값의 개수가 동일해야 한다.
- 칼럼 DEPTNO에 10, 칼럼 DNAME에는 'ACCOUNTING', 칼럼 LOC에는 'NEW YORK'을 추가

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



<실습하기> 빈 테이블에 데이터 추가하기

1. DEPT01에 데이터 추가

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

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

스크립트 출력 x 실행 결과 x

SQL | 인출된 모든 행: 1(0초)

	DEPTNO	DNAME	LOC
1	10	ACCOUNTING	NEW YORK



INSERT 구문에서 오류 발생의 예

- 칼럼 명에 기술된 목록의 수보다 VALUES 다음에 나오는 괄호 안에 기술한 값의 개수가 적으면 에러가 발생

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

- 칼럼 명에 기술된 목록의 수보다 VALUES 다음에 나오는 괄호에 작성한 값의 개수가 많으면 에러가 발생

```
INSERT INTO DEPT01  
(DEPTNO, DNAME, LOC)  
VALUES(10, 'ACCOUNTING', 'NEW YORK', 20);
```



칼럼 명을 생략한 INSERT 구문

- 테이블에 레코드를 추가할 때 **모든 칼럼에 모두 자료를 입력**하는 경우에는 굳이 칼럼 목록을 기술하지 않아도 된다.
- 칼럼 목록이 생략되면 VALUES 절 다음의 값들이 테이블의 기본 칼럼 순서대로 입력된다.
- 테이블의 컬럼 순서는 CREATE TABLE로 테이블을 생성할 때의 순서를 따른다.
- 테이블의 기본 칼럼 순서는 DESC 문의 조회 순서이다.

```
INSERT INTO DEPT01  
VALUES (20, 'RESEARCH', 'DALLAS');
```




<실습하기> 컬럼 명을 생략한 INSERT 구문 사용하기

- 컬럼명을 생략한 채 테이블이 갖은 모든 컬럼에 데이터를 추가

```
INSERT INTO DEPT01  
VALUES (20, 'RESEARCH', 'DALLAS');
```

```
INSERT INTO DEPT01
VALUES (20, 'RESEARCH', 'DALLAS');

SELECT * FROM DEPT01;
```

스크립트 출력 x

질의 결과 x

SQL | 인출된 모든 행: 2(0초)

	DEPTNO	DNAME	LOC
1	10	ACCOUNTING	NEW YORK
2	20	RESEARCH	DALLAS



<문제1>

1) EMP 테이블의 4개 칼럼의 구조를 이용하여 SAM01 테이블을 생성하시오.

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

SQL> SELECT * FROM SAM01;

선택된 레코드가 없습니다.

SQL> DESC SAM01;
이름                                널?       유형
-----
EMPNO                                NUMBER(4)
ENAME                                VARCHAR2(10)
JOB                                  VARCHAR2(9)
SAL                                  NUMBER(7,2)

SQL>
```

2) SAM01 테이블에 다음과 같은 데이터를 추가하시오.

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

SQL> SELECT * FROM SAM01;

EMPNO  ENAME      JOB          SAL
-----
1000   APPLE      POLICE       10000
1010   BANANA     NURSE        15000
1020   ORANGE     DOCTOR       25000

SQL>
```



NULL 값 삽입하는 다양한 방법

- 데이터를 입력하는 시점에서 해당 컬럼 값을 모르거나 확정되지 않았을 경우에는 NULL값을 입력해야 한다.
- NULL 값 삽입은 암시적인 방법과 명시적인 방법이 있다.
- 암시적 방법은 컬럼 명 리스트에 컬럼을 생략하는 것이다. 즉, 다른 컬럼은 값을 입력하지만 이렇게 생략한 컬럼에는 암시적으로 NULL 값이 할당되는 것입니다.
- 명시적 방법은 VALUES 리스트에 명시적으로 NULL을 입력



NULL 값 삽입하는 다양한 방법

- DEPT 테이블의 DEPTNO 컬럼은 NOT NULL 제약조건이 지정

```
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>
```

- NOT NULL 제약조건이 지정된 DEPTNO 컬럼은 널 값 입력 불가
- DEPT 테이블의 DEPTNO 컬럼에 NULL 값을 허용하지 못하도록 컬럼에 제약조건을 지정해 놓은 상태



암시적으로 NULL 값의 삽입

- 지역명이 결정되지 않은 30번 부서에 부서명만 입력
- 저장할 값을 명확하게 알고 있는 컬럼 명만 명시적으로 기술한 후에 그에 해당하는 값을 VALUES 절 다음에 기술

```
INSERT INTO DEPT01  
(DEPTNO, DNAME)  
VALUES (30, 'SALES');
```

SELECT * FROM DEPT01;

스크립트 출력 x 실행... x

SQL | 인출된 모든 행: 3(0초)

	DEPTNO	DNAME	LOC
1	10	ACCOUNTING	NEW YORK
2	20	RESEARCH	DALLAS
3	30	SALES	(null)



명시적으로 NULL 값의 삽입

- 컬럼명을 명시적으로 기술하지 않으면 테이블이 갖고 있는 모든 컬럼에 값을 지정

```
INSERT INTO DEPT01  
VALUES (40, 'OPERATIONS', NULL);
```

- 지역명이 결정되어 지지 않았더라도 반드시 값을 3개 지정해야 하기 때문에 명시적으로 VALUES 리스트에서 지역명에 NULL 입력



명시적으로 NULL 값의 삽입

- NULL 값을 갖는 칼럼을 추가하기 위해서 NULL 대신 ''를 사용할 수 있다.
- 이번에는 지역명이 아닌 부서명이 결정되지 않아 부서명에 NULL 값을 입력한 예

```
INSERT INTO DEPT01  
VALUES (50, '', 'CHICAGO');
```

	DEPTNO	DNAME	LOC
1	10	ACCOUNTING	NEW YORK
2	20	RESEARCH	DALLAS
3	30	SALES	(null)
4	50	(null)	CHICAGO



<문제2>

2. 문제 1에서 생성한 SAM01 테이블에 다음과 같이 NULL 값을 갖는 행을 추가하시오.

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

SQL> SELECT * FROM SAM01;

      EMPNO ENAME      JOB              SAL
-----
      1000 APPLE       POLICE          10000
      1010 BANANA      NURSE           15000
      1020 ORANGE      DOCTOR          25000
      1030 VERY
      1040 CAT
```




서브 쿼리로 데이터 삽입하기

- INSERT INTO 다음에 VALUES 절을 사용하는 대신에 서브 쿼리를 사용할 수 있다.
- 기존의 테이블에 있던 여러 행을 복사해서 다른 테이블에 삽입할 수 있다.
- 이 때 주의할 점은 INSERT 명령문에서 지정한 컬럼의 개수나 데이터 타입이 서브 쿼리를 수행한 결과와 동일해야 한다.



<실습하기> 서브 쿼리로 데이터 삽입하는 예제

1. 서브 쿼리로 데이터를 삽입하기 위해서 우선 테이블을 생성하되 데이터는 복사하지 않고 빈 테이블만 생성

```
DROP TABLE DEPT02;  
CREATE TABLE DEPT02  
AS  
SELECT * FROM DEPT WHERE 1=0;
```

2. 테이블 구조만을 복사해서 내용을 갖지 않는 테이블에 서브 쿼리로 레코드를 삽입한다.

```
INSERT INTO DEPT02  
SELECT * FROM DEPT;
```



<문제3>

3. 문제 1에서 생성한 SAM01 테이블에 서브 쿼리문을 사용하여 EMP 에 저장된 사원 중 10번 부서 소속 사원의 정보를 추가하시오.

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

SQL> SELECT * FROM SAM01;
```

EMPNO	ENAME	JOB	SAL
1000	APPLE	POLICE	10000
1010	BANANA	NURSE	15000
1020	ORANGE	DOCTOR	25000
1030	VERY		25000
1040	CAT		2000
1030	VERY		25000
1040	CAT		2000
7782	CLARK	MANAGER	2450
7839	KING	PRESIDENT	5000
7934	MILLER	CLERK	1300

10 개의 행이 선택되었습니다.

```
SQL>
```



02. 다중 테이블에 다중 행 입력하기

- INSERT ALL문을 사용하면 서브 쿼리의 결과를 조건 없이 여러 테이블에 동시에 입력할 수 있다 .
- 사원번호, 사원명, 입사일자로 구성된 EMP_HIR 테이블과 사원번호, 사원명, 관리자(상관))로 구성된 EMP_MGR 테이블이 빈 테이블로 존재한다고 가정한다.
- 사원 테이블(EMP)에서 부서 번호가 20인 직원들을 검색하여 EMP_HIR 테이블에는 사원번호, 사원명, 입사일자를 EMP_MGR 테이블에는 사원번호, 사원명, 해당관리자(상관)를 입력하려면 어떻게 해야 할까요?
- INSERT ALL 명령문을 사용하면 두 번의 쿼리문을 수행하지 않고도 하나의 쿼리문으로 두 개의 테이블에 원하는 컬럼 값을 삽입할 수 있다.



02. 다중 테이블에 다중 행 입력하기

- INSERT ALL 명령문은 서브 쿼리의 결과 집합을 조건 없이 여러 테이블에 동시에 입력하기 위한 명령문
- 이때 주의할 점은 서브 쿼리의 컬럼명과 데이터가 입력되는 테이블의 컬럼명이 동일해야 한다.

```
INSERT ALL  
  INTO EMP_HIR VALUES(EMPNO, ENAME, HIREDATE)  
  INTO EMP_MGR VALUES(EMPNO, ENAME, MGR)  
  SELECT EMPNO, ENAME, HIREDATE, MGR  
  FROM EMP  
 WHERE DEPTNO=20;
```



02. 다중 테이블에 다중 행 입력하기

1. 서브 쿼리로 빈 테이블 생성

```
CREATE TABLE EMP_HIR
```

```
AS
```

```
SELECT EMPNO, ENAME, HIREDATE FROM EMP WHERE 1=0;
```

```
CREATE TABLE EMP_MGR
```

```
AS
```

```
SELECT EMPNO, ENAME, MGR FROM EMP WHERE 1=0;
```

2. INSERT ALL과 서브쿼리로 레코드 삽입

```
INSERT ALL
```

```
INTO EMP_HIR VALUES(EMPNO, ENAME, HIREDATE)
```

```
INTO EMP_MGR VALUES(EMPNO, ENAME, MGR)
```

```
SELECT EMPNO, ENAME, HIREDATE, MGR
```

```
FROM EMP
```

```
WHERE DEPTNO=20;
```

```
SELECT * FROM EMP_HIR;
```

```
SELECT * FROM EMP_MGR;
```

	EMPNO	ENAME	HIREDATE
1	7369	SMITH	80/12/17
2	7566	JONES	81/04/02
3	7788	SCOTT	87/04/19
4	7876	ADAMS	87/05/23
5	7902	FORD	81/12/03

	EMPNO	ENAME	MGR
1	7369	SMITH	7902
2	7566	JONES	7839
3	7788	SCOTT	7566
4	7876	ADAMS	7788
5	7902	FORD	7566



03. 테이블 내용 수정 - UPDATE 문

- UPDATE 문은 테이블에 저장된 데이터를 수정하기 위해서 사용

```
UPDATE table_name  
SET column_name1 = value1, column_name2 = value2, ...  
WHERE conditions;
```

- UPDATE 문은 기존의 행을 수정하는 것입니다. 따라서 어떤 행의 데이터를 수정하는지 WHERE 절을 이용하여 조건을 지정
- WHERE 절을 사용하지 않을 경우는 테이블에 있는 모든 행이 수정
- 정말 테이블의 전체 행을 수정하려고 했던 것이 아니라면 큰 문제가 발생하므로 WHERE 절의 사용 유무를 신중히 판단



<실습하기> 테이블의 모든 행 변경

1. 모든 사원의 부서번호를 30번으로 수정

EMP01
테이블 생성

```
UPDATE EMP01  
SET DEPTNO=30;
```

```
CREATE TABLE EMP01  
AS  
SELECT * FROM EMP;
```

2. 모든 사원의 급여를 10% 인상시키는 UPDATE 문

```
UPDATE EMP01  
SET SAL = SAL * 1.1;
```

3. 모든 사원의 입사일을 오늘로 수정

```
UPDATE EMP01  
SET HIREDATE = SYSDATE;
```




테이블의 특정 행만 변경

- UPDATE 문에 WHERE 절을 추가하면 테이블의 특정 행이 변경
- UPDATE 문을 이용하여 테이블의 특정 행을 변경하기 위한 실습을 하기에 앞서서 실습에 사용할 테이블을 먼저 만들자.
- 이전 실습을 위해서 사용하였던 사원 테이블(EMP01)을 제거한 후 다시 기존에 있던 사원 테이블(EMP)과 동일한 구조와 데이터를 갖는 사원 테이블(EMP01)을 생성



<실습하기> 테이블의 특정 행만 변경

1. 부서번호가 10번인 사원의 부서번호를 30번으로 수정

```
UPDATE EMP01  
SET DEPTNO=30  
WHERE DEPTNO=10;
```

2. 급여가 3000 이상인 사원만 급여를 10% 인상

```
UPDATE EMP01  
SET SAL = SAL * 1.1  
WHERE SAL >= 3000;
```



<실습하기> 테이블의 특정 행만 변경

3. 1987년에 입사한 사원의 입사일을 오늘로 수정한다.
사원의 입사일을 오늘로 수정한 후에 테이블 내용 보기

```
UPDATE EMP01  
SET HIREDATE = SYSDATE  
WHERE SUBSTR(HIREDATE, 1, 2)='87';
```



<문제4>

4. SAM01 테이블에 저장된 사원 중 급여가 10000 이상인 사원들의 급여만 5000원씩 삭감하시오.

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

SQL> SELECT * FROM SAM01;
```

EMPNO	ENAME	JOB	SAL
1000	APPLE	POLICE	10000
1010	BANANA	NURSE	15000
1020	ORANGE	DOCTOR	25000
1030	VERY		25000
1040	CAT		2000
1030	VERY		25000
1040	CAT		2000
7782	CLARK	MANAGER	2450
7839	KING	PRESIDENT	5000
7934	MILLER	CLERK	1300

10 개의 행이 선택되었습니다.

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

SQL> SELECT * FROM SAM01;
```

EMPNO	ENAME	JOB	SAL
1000	APPLE	POLICE	5000
1010	BANANA	NURSE	10000
1020	ORANGE	DOCTOR	20000
1030	VERY		20000
1040	CAT		2000
1030	VERY		20000
1040	CAT		2000
7782	CLARK	MANAGER	2450
7839	KING	PRESIDENT	5000
7934	MILLER	CLERK	1300

10 개의 행이 선택되었습니다.



테이블에서 2개 이상의 칼럼 값 변경

- 테이블에서 하나의 칼럼이 아닌 복수 개 칼럼의 값을 변경하려면 기존 SET 절에 콤마를 추가하고 칼럼=값을 추가 기술하면 된다.



<실습하기> 테이블에서 2개 이상의 칼럼 값 변경

1. SCOTT 사원의 부서번호 20번, 직급은 MANAGER로 한꺼번에 수정

```
UPDATE EMP01  
SET DEPTNO=20, JOB='MANAGER'  
WHERE ENAME='SCOTT';
```

2. SCOTT 사원의 입사일자는 오늘, 급여를 50, 커미션을 4000 수정

```
UPDATE EMP01  
SET HIREDATE = SYSDATE, SAL=50, COMM=4000  
WHERE ENAME='SCOTT';
```



서브 쿼리를 이용한 데이터 수정하기

- UPDATE 문의 SET 절에서 서브 쿼리를 기술하면 서브 쿼리를 수행한 결과로 내용 변경
- 다른 테이블에 저장된 데이터로 해당 컬럼 값을 변경



<실습하기> 서브 쿼리를 이용한 데이터 수정하기

1. 20번 부서의 지역명을 10번 부서의 지역명으로 변경하기 위해서 서브 쿼리문을 사용

```
UPDATE DEPT01  
SET LOC=(SELECT LOC  
          FROM DEPT01  
          WHERE DEPTNO=10)  
WHERE DEPTNO=20;
```

	DEPTNO	DNAME	LOC
1	10	ACCOUNTING	NEW YORK
2	20	RESEARCH	DALLAS
3	30	SALES	(null)
4	50	(null)	CHICAGO



	DEPTNO	DNAME	LOC
1	10	ACCOUNTING	NEW YORK
2	20	RESEARCH	NEW YORK
3	30	SALES	(null)
4	50	(null)	CHICAGO



<문제5>

5. 서브 쿼리문을 사용하여 EMP 테이블에 저장된 데이터의 특정 컬럼만으로 구성된 SAM02 테이블을 생성하시오.

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

SQL> DESC SAM02
이름                널?       유형
-----
ENAME              VARCHAR2(10)
SAL                NUMBER(7,2)
HIREDATE           DATE
DEPTNO            NUMBER(2)

SQL>
```



<문제6>

6. 생성 후 DALLAS 에 위치한 부서 소속 직원들의 급여를 1000 인상하시오.

● [변경 전]

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

```
SQL> SELECT * FROM SAM02;
```

ENAME	SAL	HIREDATE	DEPTNO
SMITH	800	80/12/17	20
ALLEN	1600	81/02/20	30
WARD	1250	81/02/22	30
JONES	2975	81/04/02	20
MARTIN	1250	81/09/28	30
BLAKE	2850	81/05/01	30
CLARK	2450	81/06/09	10
SCOTT	3000	87/04/19	20
KING	5000	81/11/17	10
TURNER	1500	81/09/08	30
ADAMS	1100	87/05/23	20
JAMES	950	81/12/03	30
FORD	3000	81/12/03	20
MILLER	1300	82/01/23	10

14 개의 행이 선택되었습니다.

SQL>

[변경 후]

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

```
SQL> SELECT * FROM SAM02;
```

ENAME	SAL	HIREDATE	DEPTNO
SMITH	1800	80/12/17	20
ALLEN	1600	81/02/20	30
WARD	1250	81/02/22	30
JONES	3975	81/04/02	20
MARTIN	1250	81/09/28	30
BLAKE	2850	81/05/01	30
CLARK	2450	81/06/09	10
SCOTT	4000	87/04/19	20
KING	5000	81/11/17	10
TURNER	1500	81/09/08	30
ADAMS	2100	87/05/23	20
JAMES	950	81/12/03	30
FORD	4000	81/12/03	20
MILLER	1300	82/01/23	10

14 개의 행이 선택되었습니다.

SQL>



서브 쿼리를 이용한 두 개 이상의 칼럼에 대한 값 변경

- 서브 쿼리를 사용한 UPDATE 형식은 2가지(형식 2가 편리)
(형식 1)

```
UPDATE table_name  
SET column_name1 = (sub_query1),  
    column_name2 = (sub_query2), ...  
WHERE 조건
```

(형식 2)

```
UPDATE table_name  
SET (column_name1, column_name2, ...) =  
    (sub_query)  
WHERE 조건
```



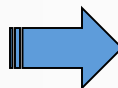


<실습하기> 서브 쿼리를 이용 한꺼번에 두 개의 컬럼 값 변경

1. 부서 번호가 50번인 부서의 이름과 지역은 null과 CHICAGO이다.
다음은 부서번호가 50인 부서의 부서명과 지역명을 부서 번호가 10번인 부서와 동일하게 변경하기 위한 UPDATE 명령문

```
UPDATE DEPT01  
SET (DNAME, LOC)=(SELECT DNAME, LOC  
                   FROM DEPT01  
                   WHERE DEPTNO=10)  
WHERE DEPTNO=50;
```

	DEPTNO	DNAME	LOC
1	10	ACCOUNTING	NEW YORK
2	20	RESEARCH	DALLAS
3	30	SALES	(null)
4	50	(null)	CHICAGO



	DEPTNO	DNAME	LOC
1	10	ACCOUNTING	NEW YORK
2	20	RESEARCH	DALLAS
3	30	SALES	(null)
4	50	ACCOUNTING	NEW YORK



<실습7>

7. 서브 쿼리문을 사용하여 SAM02 테이블의 모든 사원의 급여와 입사일을 이름이 KING 인 사원의 급여와 입사일로 변경하시오.

● [변경 전]

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

```
SQL> SELECT * FROM SAM02;
```

ENAME	SAL	HIREDATE	DEPTNO
SMITH	1800	80/12/17	20
ALLEN	1600	81/02/20	30
WARD	1250	81/02/22	30
JONES	3975	81/04/02	20
MARTIN	1250	81/09/28	30
BLAKE	2850	81/05/01	30
CLARK	2450	81/06/09	10
SCOTT	4000	87/04/19	20
KING	5000	81/11/17	10
TURNER	1500	81/09/08	30
ADAMS	2100	87/05/23	20
JAMES	950	81/12/03	30
FORD	4000	81/12/03	20
MILLER	1300	82/01/23	10

14 개의 행이 선택되었습니다.

SQL>

[변경 후]

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

```
SQL> SELECT * FROM SAM02;
```

ENAME	SAL	HIREDATE	DEPTNO
SMITH	5000	81/11/17	20
ALLEN	5000	81/11/17	30
WARD	5000	81/11/17	30
JONES	5000	81/11/17	20
MARTIN	5000	81/11/17	30
BLAKE	5000	81/11/17	30
CLARK	5000	81/11/17	10
SCOTT	5000	81/11/17	20
KING	5000	81/11/17	10
TURNER	5000	81/11/17	30
ADAMS	5000	81/11/17	20
JAMES	5000	81/11/17	30
FORD	5000	81/11/17	20
MILLER	5000	81/11/17	10

14 개의 행이 선택되었습니다.

SQL>



04. 테이블에 불필요한 행을 삭제 DELETE 문

- DELETE 문은 테이블에 저장되어 있는 데이터를 삭제

```
DELETE FROM table_name  
WHERE conditions;
```

- DELETE 문은 테이블의 기존 행을 삭제하며 특정한 로우(행)을 삭제하기 위해서 WHERE 절을 이용하여 조건을 지정
- 만약 DELETE 문에 WHERE 절을 사용하지 않을 경우 테이블에 있는 모든 행이 삭제되므로 매우 신중하게 명령문을 사용





<실습하기> 조건을 제시하여 특정 행만 삭제하기

- 부서 테이블에서 30번 부서만 삭제

```
DELETE FROM DEPT01  
WHERE DEPTNO=30;
```

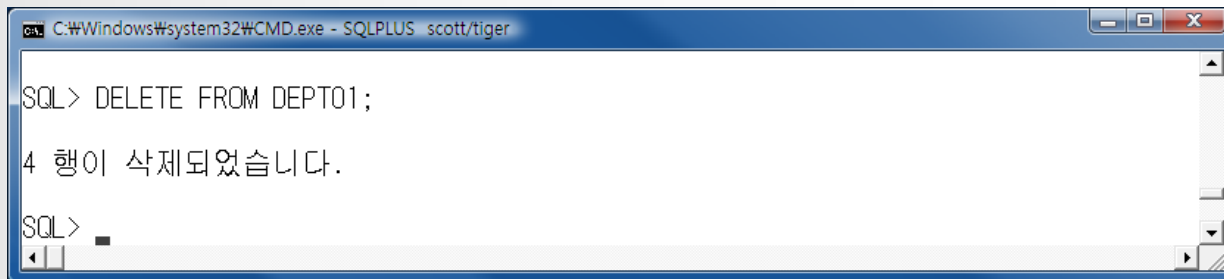
```
C:\Windows\system32\CMD.exe - SQLPLUS scott/tiger  
  
SQL> DELETE FROM DEPT01  
2  WHERE DEPTNO=30;  
  
1 행이 삭제되었습니다.  
  
SQL> SELECT * FROM DEPT01;  
  
DEPTNO DNAME      LOC  
-----  
10 ACCOUNTING   NEW YORK  
20 RESEARCH     DALLAS  
40 OPERATIONS   BOSTON  
  
SQL>
```



<실습하기> DELETE 문으로 행을 삭제하기

- DELETE 문으로 부서 테이블의 모든 행을 삭제

DELETE FROM DEPT01;



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

SQL> DELETE FROM DEPT01;

4 행이 삭제되었습니다.

SQL>
```




<실습8>

8. SAM01 테이블에서 직책이 정해지지 않은 사원을 삭제하시오.

[삭제 전]

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

```
SQL> SELECT * FROM SAM01;
```

EMPNO	ENAME	JOB	SAL
1000	APPLE	POLICE	5000
1010	BANANA	NURSE	10000
1020	ORANGE	DOCTOR	20000
1030	VERY		20000
1040	CAT		2000
1030	VERY		20000
1040	CAT		2000
7782	CLARK	MANAGER	2450
7839	KING	PRESIDENT	5000
7934	MILLER	CLERK	1300

10 개의 행이 선택되었습니다.

[삭제 후]

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

```
SQL> SELECT * FROM SAM01;
```

EMPNO	ENAME	JOB	SAL
1000	APPLE	POLICE	5000
1010	BANANA	NURSE	10000
1020	ORANGE	DOCTOR	20000
7782	CLARK	MANAGER	2450
7839	KING	PRESIDENT	5000
7934	MILLER	CLERK	1300

6 개의 행이 선택되었습니다.

SQL>



서브 쿼리를 이용한 데이터 삭제

- DELETE 문을 사용하기에 앞서 사원 테이블을 복사한다. 사원 테이블에서 부서명이 SALES인 사원을 모두 삭제해보도록 한다.
- 사원 테이블에는 부서명이 기록되어 있지 않다.
- 부서명은 부서 테이블에 기록되어 있으므로 부서 테이블에서 부서명이 SALES인 부서의 번호부터 알아내야 한다.
- 이렇게 알아낸 부서번호를 사원 테이블에 적용하기 위해서는 서브 쿼리를 이용해야 한다.





<실습하기> 서브 쿼리를 이용한 데이터 삭제하기

- 사원 테이블에서 부서명이 SALES인 사원을 모두 삭제

```
DELETE FROM EMP01  
WHERE DEPTNO=(SELECT DEPTNO  
               FROM DEPT  
               WHERE DNAME='SALES');
```

