Advanced Product Service

Oracle Database 11g : SQL

Creation Date	:	2013년 05월 05일
Last Updated	:	2013년 05월 14일
Version	:	1.0

Chong Ha, Ryu

chongha.ryu@gmail.com
http://oukr.tistory.com

1. SQL SELECT 문을 사용하여 데이터 검색

[orcl:~]$ sqlplus   ora1/oracle

- *현재 유저가 가지고 있는 TABLE, VIEW 의 목록*

SQL> SELECT * FROM tab ;

```
TNAME                          TABTYPE  CLUSTERID
------------------------------ ------- ----------
BONUS                          TABLE
COUNTRIES                      TABLE
DEPARTMENTS                    TABLE
DEPT                           TABLE
EMP                            TABLE
EMPLOYEES                      TABLE
EMP_DETAILS_VIEW               VIEW
JOBS                           TABLE
JOB_GRADES                     TABLE
JOB_HISTORY                    TABLE
LOCATIONS                      TABLE
REGIONS                        TABLE
SALGRADE                       TABLE
```

- *테이블의 구조 확인*

SQL> DESCRIBE   DEPT

```
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DEPTNO                                             NUMBER(2)
 DNAME                                              VARCHAR2(14)
 LOC                                                VARCHAR2(13)
```

SQL> DESC   EMP (※ 약어 사용 가능)

```
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPNO                                              NUMBER(4)
 ENAME                                              VARCHAR2(10)
 JOB                                                VARCHAR2(9)
 MGR                                                NUMBER(4)
 HIREDATE                                           DATE
 SAL                                                NUMBER(7,2)
 COMM                                               NUMBER(7,2)
 DEPTNO                                             NUMBER(2)
```

- *테이블의 모든 컬럼의 데이터 검색*

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 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
...
```

- *테이블에서 특정 컬럼의 데이터 검색*

```
SQL> SELECT deptno, dname FROM dept ;
    DEPTNO DNAME
---------- --------------
        10 ACCOUNTING
        20 RESEARCH
        30 SALES
        40 OPERATIONS
SQL> SELECT empno, ename, deptno FROM emp ;
     EMPNO ENAME          DEPTNO
---------- ---------- ----------
      7369 SMITH              20
      7499 ALLEN              30
      7521 WARD               30
...
```

- *리터럴 값 사용*

```
SQL> SELECT empno, ename, 'RYU', 100, '2013/05/05' FROM emp ;
     EMPNO ENAME      'RY        100 '2013/05/0
---------- ---------- --- ---------- ----------
      7369 SMITH      RYU        100 2013/05/05
      7499 ALLEN      RYU        100 2013/05/05
      7521 WARD       RYU        100 2013/05/05
      7566 JONES      RYU        100 2013/05/05
...
```

- *산술식 사용*

```
SQL> SELECT empno, ename, sal * 12, deptno
       FROM emp ;
     EMPNO ENAME          SAL*12     DEPTNO
---------- ---------- ---------- ----------
      7369 SMITH            9600         20
      7499 ALLEN           19200         30
      7521 WARD            15000         30
...
SQL> SELECT empno, ename, hiredate, hiredate + 1
       FROM emp ;
     EMPNO ENAME      HIREDATE  HIREDATE+
---------- ---------- --------- ---------
      7369 SMITH      17-DEC-80 18-DEC-80
      7499 ALLEN      20-FEB-81 21-FEB-81
      7521 WARD       22-FEB-81 23-FEB-81
...
```

- *Data Type 에 따라 산술식 사용 불가능한 경우 존재*

```
SQL> SELECT empno, ename + 'A' FROM emp ;
SELECT empno, ename + 'A' FROM emp
              *
ERROR at line 1:
ORA-01722: invalid number
```

- *NULL 값 확인*

```
SQL> SELECT empno, ename, sal, comm, mgr FROM emp ;
     EMPNO ENAME             SAL       COMM        MGR
---------- ---------- ---------- ---------- ----------
      7369 SMITH             800 (NULL)           7902
      7499 ALLEN            1600        300       7698
      7521 WARD             1250        500       7698
      7566 JONES            2975 (NULL)           7839
      7654 MARTIN           1250       1400       7698
      7698 BLAKE            2850 (NULL)           7839
      7782 CLARK            2450 (NULL)           7839
      7788 SCOTT            3000 (NULL)           7566
      7839 KING             5000 (NULL)     (NULL)
...
```

```
SQL> SELECT empno, sal, comm, sal+comm FROM emp ;
     EMPNO        SAL       COMM   SAL+COMM
---------- ---------- ---------- ----------
      7369        800 (NULL)     (NULL)
      7499       1600        300       1900
      7521       1250        500       1750
      7566       2975 (NULL)     (NULL)
...
```

```
SQL> SELECT empno, sal, comm, sal + NVL(comm,0)
       FROM emp ;
     EMPNO        SAL       COMM SAL+NVL(COMM,0)
---------- ---------- ---------- ---------------
      7369        800 (NULL)                 800
      7499       1600        300            1900
      7521       1250        500            1750
      7566       2975 (NULL)                2975
...
```

- *컬럼 별칭 (Alias) 사용*

```
SQL> SELECT empno   id, ename AS name, sal*12   "Annual Salary"
       FROM emp ;
        ID NAME       Annual Salary
---------- ---------- -------------
      7369 SMITH               9600
      7499 ALLEN              19200
      7521 WARD               15000
      7566 JONES              35700
...
```

- *연결 연산자 사용*

```
SQL> SELECT ename || job FROM emp ;
ENAME||JOB
-------------------
SMITHCLERK
ALLENSALESMAN
WARDSALESMAN
JONESMANAGER
...
```

```
SQL> SELECT ename || ' is a ' || job FROM emp ;
ENAME||'ISA'||JOB
-------------------------
SMITH is a CLERK
ALLEN is a SALESMAN
WARD is a SALESMAN
JONES is a MANAGER
...
SQL> SELECT ename || ''s Salary:' || sal FROM emp ;
ERROR:
ORA-01756: quoted string not properly terminated
SQL> SELECT ename || '''s Salary:' || sal FROM emp ;
ENAME||'''SSALARY:'||SAL
------------------------------------------------------------
SMITH's Salary:800
ALLEN's Salary:1600
WARD's Salary:1250
JONES's Salary:2975
...
SQL> SELECT ename || q'['s Salary:]'|| sal FROM emp ;
ENAME||Q'['SSALARY:]'||SAL
------------------------------------------------------------
SMITH's Salary:800
ALLEN's Salary:1600
WARD's Salary:1250
JONES's Salary:2975
...
```

- *중복 행 제거*

```
SQL> SELECT mgr FROM emp ;
       MGR
----------
      7902
      7698
      7698
      7839
      7698
      7839
      7839
      7566
(NULL)
      7698
      7788
      7698
      7566
      7782
SQL> SELECT DISTINCT mgr FROM emp ;
       MGR
----------
      7839
(NULL)
      7782
      7698
      7902
      7566
      7788
```

```
SQL> SELECT deptno FROM emp ;
    DEPTNO
----------
        20
        30
        30
        20
        30
        30
        10
        20
        10
        30
        20
        30
        20
        10
SQL> SELECT DISTINCT deptno FROM emp ;
    DEPTNO
----------
        30
        20
        10
```

Quiz

1. EMPLOYEES, DEPARTMENTS 테이블의 모든 컬럼의 데이터를 검색 하시오.

2. EMPLOYEES 테이블에서 사원의 이름, 급여, 부서번호 컬럼의 데이터를 검색 하시오.

3. EMPLOYEES 테이블에서 각 사원의 이름 및 실제 커미션을 검색 하시오. (NULL 의 경우 0을 표시)

```
FIRST_NAME           LAST_NAME                     SALARY COMMISSION_PCT       COMM
-------------------- ------------------------- ---------- -------------- ----------
...
Randall              Matos                           2600 (NULL)                  0
Peter                Vargas                          2500 (NULL)                  0
Eleni                Zlotkey                        10500             .2       2100
Ellen                Abel                           11000             .3       3300
Jonathon             Taylor                          8600             .2       1720
Kimberely            Grant                           7000            .15       1050
Jennifer             Whalen                          4400 (NULL)                  0
...
```

2. 데이터 제한 및 정렬

- *WHERE 절의 사용*

```
SQL> SELECT empno, ename, sal, hiredate, deptno
       FROM emp
       WHERE deptno = 30 ;
     EMPNO ENAME             SAL HIREDATE      DEPTNO
---------- ---------- ---------- --------- ----------
      7499 ALLEN            1600 20-FEB-81         30
      7521 WARD             1250 22-FEB-81         30
      7654 MARTIN           1250 28-SEP-81         30
      7698 BLAKE            2850 01-MAY-81         30
      7844 TURNER           1500 08-SEP-81         30
      7900 JAMES             950 03-DEC-81         30
```

```
SQL> SELECT empno, ename, sal, hiredate, deptno
       FROM emp
       WHERE ename = 'SCOTT' ;
     EMPNO ENAME             SAL HIREDATE      DEPTNO
---------- ---------- ---------- --------- ----------
      7788 SCOTT            3000 09-DEC-82         20
```

```
SQL> SELECT empno, ename, sal, hiredate, deptno
       FROM emp
       WHERE ename = SCOTT ;
WHERE ename = SCOTT
              *
ERROR at line 3:
ORA-00904: "SCOTT": invalid identifier
```

```
SQL> SELECT empno, ename, sal, hiredate, deptno
       FROM emp
       WHERE ename = 'scott' ;
no rows selected
```

```
SQL> SELECT empno, ename, sal, hiredate, deptno
       FROM emp
       WHERE hiredate = '81/12/03' ;
WHERE hiredate = '81/12/03'
                  *
ERROR at line 3:
ORA-01847: day of month must be between 1 and last day of month
```

```
SQL> SELECT empno, ename, sal, hiredate, deptno
       FROM emp
       WHERE hiredate = '03-DEC-81' ;
EMPNO ENAME             SAL HIREDATE      DEPTNO
---------- ---------- ---------- --------- ----------
      7900 JAMES             950 03-DEC-81         30
      7902 FORD             3000 03-DEC-81         20
```

※ 문자,날짜의 비교는 작은 따옴표로 묶어야 하며 문자는 대소문자, 날짜는 형식이 동일해야 검색 가능

- *단일 행 비교 연산자 사용*

```
SQL> SELECT ename, sal, hiredate, deptno
       FROM emp
       WHERE sal >= 3000 ;
ENAME             SAL HIREDATE      DEPTNO
---------- ---------- --------- ----------
SCOTT            3000 09-DEC-82         20
KING             5000 17-NOV-81         10
FORD             3000 03-DEC-81         20
SQL> SELECT ename, sal, hiredate, deptno
       FROM emp
       WHERE deptno != 30 ;
ENAME             SAL HIREDATE      DEPTNO
---------- ---------- --------- ----------
SMITH             800 17-DEC-80         20
JONES            2975 02-APR-81         20
CLARK            2450 09-JUN-81         10
SCOTT            3000 09-DEC-82         20
KING             5000 17-NOV-81         10
ADAMS            1100 12-JAN-83         20
FORD             3000 03-DEC-81         20
MILLER           1300 23-JAN-82         10
SQL> SELECT ename, sal, hiredate, deptno
       FROM emp
       WHERE ename >= 'SCOTT' ;
ENAME             SAL HIREDATE      DEPTNO
---------- ---------- --------- ----------
SMITH             800 17-DEC-80         20
WARD             1250 22-FEB-81         30
SCOTT            3000 09-DEC-82         20
TURNER           1500 08-SEP-81         30
SQL> SELECT ename, sal, hiredate, deptno
       FROM emp
       WHERE hiredate >= '31-DEC-81' ;     (또는 WHERE hiredate >= '81/12/31')
ENAME             SAL HIREDATE      DEPTNO
---------- ---------- --------- ----------
SCOTT            3000 09-DEC-82         20
ADAMS            1100 12-JAN-83         20
MILLER           1300 23-JAN-82         10
```

- *BETWEEN 연산자 사용*

```
SQL> SELECT ename, sal, hiredate, deptno
       FROM emp
       WHERE sal BETWEEN 2000 AND 3000 ;
ENAME             SAL HIREDATE      DEPTNO
---------- ---------- --------- ----------
JONES            2975 02-APR-81         20
BLAKE            2850 01-MAY-81         30
CLARK            2450 09-JUN-81         10
SCOTT            3000 09-DEC-82         20
FORD             3000 03-DEC-81         20
```

```
SQL> SELECT ename, sal, hiredate, deptno
     FROM emp
     WHERE sal >= 2000 AND sal <= 3000 ;
ENAME             SAL HIREDATE      DEPTNO
---------- ---------- --------- ----------
JONES            2975 02-APR-81         20
BLAKE            2850 01-MAY-81         30
CLARK            2450 09-JUN-81         10
SCOTT            3000 09-DEC-82         20
FORD             3000 03-DEC-81         20
SQL> SELECT ename, sal, hiredate, deptno
     FROM emp
     WHERE ename BETWEEN 'ADAMS' AND 'CLARK' ;
ENAME             SAL HIREDATE      DEPTNO
---------- ---------- --------- ----------
ALLEN            1600 20-FEB-81         30
BLAKE            2850 01-MAY-81         30
CLARK            2450 09-JUN-81         10
ADAMS            1100 12-JAN-83         20
SQL> SELECT ename, sal, hiredate, deptno
     FROM emp
     WHERE hiredate BETWEEN '01-JAN-82' AND '31-DEC-82' ;
ENAME             SAL HIREDATE      DEPTNO
---------- ---------- --------- ----------
SCOTT            3000 09-DEC-82         20
MILLER           1300 23-JAN-82         10
```

- *IN 연산자 사용*

```
SQL> SELECT ename, sal, deptno
     FROM emp
     WHERE deptno IN (10,20) ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
SMITH             800         20
JONES            2975         20
CLARK            2450         10
SCOTT            3000         20
KING             5000         10
ADAMS            1100         20
FORD             3000         20
MILLER           1300         10
SQL> SELECT ename, sal, deptno
     FROM emp
     WHERE deptno = 10
         OR deptno = 20 ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
SMITH             800         20
JONES            2975         20
CLARK            2450         10
SCOTT            3000         20
. . .
```

```
SQL> SELECT ename, job, deptno
       FROM emp
       WHERE job IN ('MANAGER','CLERK') ;
ENAME      JOB           DEPTNO
---------- --------- ----------
SMITH      CLERK             20
JONES      MANAGER           20
BLAKE      MANAGER           30
CLARK      MANAGER           10
ADAMS      CLERK             20
JAMES      CLERK             30
MILLER     CLERK             10
```

• LIKE 연산자 사용

```
SQL> SELECT last_name, hire_date, salary
       FROM employees
       WHERE last_name LIKE 'M%' ;
LAST_NAME                 HIRE_DATE     SALARY
------------------------- --------- ----------
Matos                     15-MAR-98       2600
Mourgos                   16-NOV-99       5800
SQL> SELECT last_name, hire_date, salary
       FROM employees
       WHERE last_name LIKE 'M____' ;
LAST_NAME                 HIRE_DATE     SALARY
------------------------- --------- ----------
Matos                     15-MAR-98       2600
SQL> SELECT last_name, job_id, salary
       FROM employees
       WHERE job_id LIKE '%A_%' ;
LAST_NAME                 JOB_ID         SALARY
------------------------- ---------- ----------
King                      AD_PRES         24000
Kochhar                   AD_VP           17000
De Haan                   AD_VP           17000
Mourgos                   ST_MAN           5800
Zlotkey                   SA_MAN          10500
Abel                      SA_REP          11000
Taylor                    SA_REP           8600
Grant                     SA_REP           7000
Whalen                    AD_ASST          4400
Hartstein                 MK_MAN          13000
Higgins                   AC_MGR          12000
Gietz                     AC_ACCOUNT       8300
SQL> SELECT last_name, job_id, salary
       FROM employees
       WHERE job_id LIKE '%A\_%'   ESCAPE '\' ;
LAST_NAME                 JOB_ID         SALARY
------------------------- ---------- ----------
Zlotkey                   SA_MAN          10500
Abel                      SA_REP          11000
Taylor                    SA_REP           8600
Grant                     SA_REP           7000
```

- *IS NULL 연산자 사용*

```
SQL> SELECT empno, sal, comm, deptno
       FROM emp
       WHERE comm IS NULL ;
     EMPNO        SAL       COMM     DEPTNO
---------- ---------- ---------- ----------
      7369        800 (NULL)             20
      7566       2975 (NULL)             20
      7698       2850 (NULL)             30
      7782       2450 (NULL)             10
      7788       3000 (NULL)             20
      7839       5000 (NULL)             10
      7876       1100 (NULL)             20
      7900        950 (NULL)             30
      7902       3000 (NULL)             20
      7934       1300 (NULL)             10
SQL> SELECT empno, sal, comm, deptno
       FROM emp
       WHERE comm = '' ;
no rows selected
```

- *AND, OR 동시 사용 시 주의 사항 확인*

```
SQL> SELECT ename, sal, deptno
       FROM emp
       WHERE deptno = 10
            OR deptno = 20
          AND sal >= 3000 ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
CLARK            2450         10
SCOTT            3000         20
KING             5000         10
FORD             3000         20
MILLER           1300         10
SQL> SELECT ename, sal, deptno
       FROM emp
       WHERE (deptno = 10
            OR deptno = 20)
          AND sal >= 3000 ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
SCOTT            3000         20
KING             5000         10
FORD             3000         20

SQL> SELECT ename, sal, deptno
       FROM emp
       WHERE deptno IN (10,20)
          AND sal >= 3000 ;
...
```

- *NOT 연산자 사용*

```
SQL> SELECT ename, sal, deptno
       FROM emp
       WHERE sal NOT BETWEEN 1000 AND 3000 ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
SMITH             800         20
KING             5000         10
JAMES             950         30
SQL> SELECT ename, sal, deptno
       FROM emp
       WHERE deptno NOT IN (20,30) ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
CLARK            2450         10
KING             5000         10
MILLER           1300         10
SQL> SELECT ename, sal, deptno
       FROM emp
       WHERE ename NOT LIKE 'A%' ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
SMITH             800         20
WARD             1250         30
JONES            2975         20
MARTIN           1250         30
...
SQL> SELECT ename, sal, comm, deptno
       FROM emp
       WHERE comm IS NOT NULL ;
ENAME             SAL       COMM     DEPTNO
---------- ---------- ---------- ----------
ALLEN            1600        300         30
WARD             1250        500         30
MARTIN           1250       1400         30
TURNER           1500          0         30
```

- *ORDER BY 절의 사용*

```
SQL> SELECT *
       FROM dept
       ORDER BY dname ;
    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        40 OPERATIONS     BOSTON
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
```

```
SQL> SELECT *
       FROM dept
       ORDER BY dname DESC ;
    DEPTNO DNAME          LOC
---------- -------------- -------------
        30 SALES          CHICAGO
        20 RESEARCH       DALLAS
        40 OPERATIONS     BOSTON
        10 ACCOUNTING     NEW YORK
SQL> SELECT *
       FROM dept
       ORDER BY 2 ;
    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        40 OPERATIONS     BOSTON
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
SQL> SELECT ename, sal, comm, deptno
       FROM emp
       ORDER BY deptno, sal DESC ;
ENAME             SAL       COMM     DEPTNO
---------- ---------- ---------- ----------
KING             5000 (NULL)             10
CLARK            2450 (NULL)             10
MILLER           1300 (NULL)             10
SCOTT            3000 (NULL)             20
FORD             3000 (NULL)             20
JONES            2975 (NULL)             20
ADAMS            1100 (NULL)             20
SMITH             800 (NULL)             20
BLAKE            2850 (NULL)             30
ALLEN            1600        300         30
TURNER           1500          0         30
MARTIN           1250       1400         30
WARD             1250        500         30
JAMES             950 (NULL)             30
SQL> SELECT ename, sal, comm, deptno
       FROM emp
       ORDER BY 4, 2 DESC ;
...
SQL> SELECT ename, sal, sal*12 ANN_SAL
       FROM emp
       ORDER BY sal * 12 ;
ENAME             SAL    ANN_SAL
---------- ---------- ----------
SMITH             800       9600
JAMES             950      11400
ADAMS            1100      13200
WARD             1250      15000
MARTIN           1250      15000
MILLER           1300      15600
TURNER           1500      18000
...
```

```
SQL> SELECT ename, sal, sal*12 ANN_SAL
       FROM emp
       ORDER BY ANN_SAL ;
...
```

- *치환변수 사용*

```
SQL> SELECT empno, ename, sal, deptno
       FROM emp
       WHERE empno = &id ;
Enter value for id: 7788
old   3:        WHERE empno = &id
new   3:        WHERE empno = 7788
     EMPNO ENAME             SAL     DEPTNO
---------- ---------- ---------- ----------
      7788 SCOTT            3000         20
SQL> ACCEPT  id  NUMBER  PROMPT  'Employee ID : '
Employee ID : 7566
SQL> SELECT empno, ename, sal, deptno
       FROM emp
       WHERE empno = &id ;
old   3:        WHERE empno = &id
new   3:        WHERE empno = 7566
     EMPNO ENAME             SAL     DEPTNO
---------- ---------- ---------- ----------
      7566 JONES            2975         20
SQL> SET VERIFY OFF
SQL> SELECT empno, ename, sal, deptno
       FROM emp
       WHERE empno = &id ;
     EMPNO ENAME             SAL     DEPTNO
---------- ---------- ---------- ----------
      7566 JONES            2975         20
SQL> DEFINE  ID
DEFINE ID              =      7566 (NUMBER)
SQL> UNDEFINE  ID
SQL> SELECT empno, ename, sal, deptno
       FROM emp
       WHERE empno = &id ;
Enter value for id: 7839
     EMPNO ENAME             SAL     DEPTNO
---------- ---------- ---------- ----------
      7839 KING             5000         10
```

Quiz

1. EMP 테이블에서 급여보다 커미션을 더 많이 받는 사원의 데이터를 검색 하시오.

```
ENAME             SAL       COMM     DEPTNO
---------- ---------- ---------- ----------
MARTIN           1250       1400         30
```

2. EMP 테이블에서 입사월이 12월인 사원의 데이터를 검색 하시오.

```
ENAME      HIREDATE      DEPTNO
---------- --------- ----------
SMITH      17-DEC-80         20
SCOTT      09-DEC-82         20
JAMES      03-DEC-81         30
FORD       03-DEC-81         20
```

3. EMP 테이블에서 부서번호 30번을 가지고 있으며 급여는 1500보다 많은 급여를 받는 사원의 데이터를 검색 하시오. 검색 시 사원의 연봉을 계산하고 소득세율을 함께 표시하며 급여를 기준으로 내림차순 정렬 한다.

연봉 : (12 * SAL) + COMM

소득세율 : 연봉 * 0.03

```
ENAME             SAL       COMM    ANN_SAL        TAX
---------- ---------- ---------- ---------- ----------
BLAKE            2850 (NULL)          34200       1026
ALLEN            1600        300      19500        585
```

3. 단일 행 함수를 사용하여 출력 커스터마이즈

- *문자함수*

```
SQL> SELECT last_name, job_id, UPPER(last_name), LOWER(last_name), INITCAP(job_id)
       FROM employees
       WHERE department_id = 20 ;
LAST_NAME                 JOB_ID     UPPER(LAST_NAME)          LOWER(LAST_NAME)          INITCAP(JO
------------------------- ---------- ------------------------- ------------------------- ----------
Hartstein                 MK_MAN     HARTSTEIN                 hartstein                 Mk_Man
Fay                       MK_REP     FAY                       fay                       Mk_Rep
```

```
SQL> SELECT ename, job, ename||job, CONCAT(ename,job)
       FROM emp
       WHERE deptno = 10 ;
ENAME      JOB       ENAME||JOB          CONCAT(ENAME,JOB)
---------- --------- ------------------- -------------------
CLARK      MANAGER   CLARKMANAGER        CLARKMANAGER
KING       PRESIDENT KINGPRESIDENT       KINGPRESIDENT
MILLER     CLERK     MILLERCLERK         MILLERCLERK
```

```
SQL> SELECT ename, SUBSTR(ename,1,3), SUBSTR(ename,3), SUBSTR(ename, -2,2)
       FROM emp
       WHERE deptno = 10 ;
ENAME      SUBSTR(ENAME SUBSTR(ENAME,3)                  SUBSTR(E
---------- ------------ -------------------------------- --------
CLARK      CLA          ARK                              RK
KING       KIN          NG                               NG
MILLER     MIL          LLER                             ER
```

```
SQL> SELECT ename, LENGTH(ename), INSTR(ename, 'L'), INSTR(ename, 'L', 1, 2)
       FROM emp
       WHERE deptno = 10 ;
ENAME      LENGTH(ENAME) INSTR(ENAME,'L') INSTR(ENAME,'L',1,2)
---------- ------------- ---------------- --------------------
CLARK                  5                2                    0
KING                   4                0                    0
MILLER                 6                3                    4
```

```
SQL> SELECT sal, LPAD(sal, 7, '0') , RPAD(sal, 7,'*')
       FROM emp
       WHERE deptno = 10 ;
       SAL LPAD(SAL,7,'0')             RPAD(SAL,7,'*')
---------- ---------------------------- ----------------------------
      2450 0002450                      2450***
      5000 0005000                      5000***
      1300 0001300                      1300***
```

```
SQL> SELECT ename, REPLACE(ename, 'CL','M')
       FROM emp
       WHERE deptno = 10 ;
ENAME      REPLACE(EN
---------- ----------
CLARK      MARK
KING       KING
MILLER     MILLER
```

```
SQL> SELECT ename, TRIM('S' FROM ename), TRIM(LEADING 'S' FROM ename), TRIM(TRAILING 'S' FROM ename)
       FROM emp
       WHERE deptno = 20 ;
ENAME      TRIM('S'FR TRIM(LEADI TRIM(TRAIL
---------- ---------- ---------- ----------
SMITH      MITH       MITH       SMITH
JONES      JONE       JONES      JONE
SCOTT      COTT       COTT       SCOTT
ADAMS      ADAM       ADAMS      ADAM
FORD       FORD       FORD       FORD
SQL> INSERT INTO emp (empno, ename, deptno)
      VALUES (1234, '   RYU', 10) ;
SQL> SELECT ename, LENGTH(ename), TRIM(ename)
        FROM emp
        WHERE deptno = 10 ;
ENAME      LENGTH(ENAME) TRIM(ENAME
---------- ------------- ----------
  RYU                  5 RYU
CLARK                  5 CLARK
KING                   4 KING
MILLER                 6 MILLER
SQL> ROLLBACK ;
```

- *숫자 함수 사용*

```
SQL> SELECT sal/7, ROUND(sal/7, 2), ROUND(sal/7, 0), ROUND(sal/7, -1)
       FROM emp
       WHERE deptno = 10 ;
     SAL/7 ROUND(SAL/7,2) ROUND(SAL/7,0) ROUND(SAL/7,-1)
---------- -------------- -------------- ---------------
       350            350            350             350
714.285714         714.29            714             710
185.714286         185.71            186             190
SQL> SELECT sal/7, TRUNC(sal/7,2), TRUNC(sal/7,0), TRUNC(sal/7, -1)
       FROM emp
       WHERE deptno = 10 ;
     SAL/7 TRUNC(SAL/7,2) TRUNC(SAL/7,0) TRUNC(SAL/7,-1)
---------- -------------- -------------- ---------------
       350            350            350             350
714.285714         714.28            714             710
185.714286         185.71            185             180

SQL> SELECT sal, MOD(sal, 1000)
       FROM emp
       WHERE deptno = 10 ;
SAL MOD(SAL,1000)
---------- -------------
      2450           450
      5000             0
      1300           300
```

- *DUAL 테이블 & SYSDATE 함수 사용*

```
SQL> DESC dual
Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DUMMY                                              VARCHAR2(1)
SQL> SELECT * FROM dual ;
D
-
X
SQL> SELECT   SYSDATE   FROM dept ;
SYSDATE
---------
06-MAY-13
06-MAY-13
06-MAY-13
06-MAY-13
SQL> SELECT   SYSDATE   FROM dual ;
SYSDATE
---------
06-MAY-13
```

- *날짜 작업*

```
SQL> ALTER SESSION SET NLS_DATE_FORMAT = 'YYYY/MM/DD HH24:MI:SS' ;
SQL> SELECT   SYSDATE   FROM dual ;
SYSDATE
-------------------
2013/05/06 00:36:34
SQL> SELECT   SYSDATE, SYSDATE + 1, SYSDATE + 5/24, SYSDATE + 5/1440, SYSDATE + 5/86400
       FROM dual ;
SYSDATE             SYSDATE+1           SYSDATE+5/24        SYSDATE+5/1440      SYSDATE+5/86400
------------------- ------------------- ------------------- ------------------- -------------------
2013/05/06 00:38:25 2013/05/07 00:38:25 2013/05/06 05:38:25 2013/05/06 00:43:25 2013/05/06 00:38:30
SQL> SELECT   ename, SYSDATE - hiredate
       FROM emp
       WHERE deptno = 10 ;
ENAME      SYSDATE-HIREDATE
---------- ----------------
CLARK             11654.028
KING              11493.028
MILLER            11426.028
```

- *RR 날짜 형식 확인*

```
SQL> CREATE TABLE copy_emp
       AS SELECT * FROM emp ;
SQL> ALTER SESSION SET NLS_DATE_FORMAT = 'RR/MM/DD' ;
SQL> UPDATE   copy_emp
       SET hiredate = '13/05/06'
       WHERE empno = 7369 ;
```

```
SQL> UPDATE   copy_emp
        SET hiredate = '99/05/06'
        WHERE empno = 7566 ;
SQL> ALTER SESSION SET NLS_DATE_FORMAT = 'YY/MM/DD' ;
SQL> UPDATE   copy_emp
        SET hiredate = '13/05/06'
        WHERE empno = 7788 ;
SQL> UPDATE   copy_emp
        SET hiredate = '99/05/06'
        WHERE empno = 7876 ;
SQL> ALTER SESSION SET NLS_DATE_FORMAT = 'YYYY/MM/DD' ;
SQL> SELECT empno, hiredate
        FROM copy_emp
        WHERE deptno = 20 ;
     EMPNO HIREDATE
---------- ----------
      7369 2013/05/06
      7566 1999/05/06
      7788 2013/05/06
      7876 2099/05/06
      7902 1981/12/03
```

- *날짜 함수 사용*

```
SQL> SELECT sysdate, hiredate, MONTHS_BETWEEN(sysdate, hiredate)
        FROM emp
        WHERE deptno = 10 ;
SYSDATE    HIREDATE   MONTHS_BETWEEN(SYSDATE,HIREDATE)
---------- ---------- --------------------------------
2013/05/06 1981/06/09                       382.904391
2013/05/06 1981/11/17                       377.646327
2013/05/06 1982/01/23                       375.452779
SQL> SELECT sysdate, ADD_MONTHS(sysdate, 3),
                ADD_MONTHS('2012/02/28',3), ADD_MONTHS('2012/02/29',3)
        FROM dual ;
SYSDATE    ADD_MONTHS ADD_MONTHS ADD_MONTHS
---------- ---------- ---------- ----------
2013/05/06 2013/08/06 2012/05/28 2012/05/31

SQL> SELECT hiredate, NEXT_DAY(hiredate, 'MONDAY'), LAST_DAY(hiredate)
        FROM emp
        WHERE deptno = 10 ;
HIREDATE   NEXT_DAY(H LAST_DAY(H
---------- ---------- ----------
1981/06/09 1981/06/15 1981/06/30
1981/11/17 1981/11/23 1981/11/30
1982/01/23 1982/01/25 1982/01/31
```

```
SQL> SELECT hiredate, ROUND(hiredate,'MONTH'), ROUND(hiredate,'YEAR')
       FROM emp
       WHERE deptno = 10 ;
HIREDATE   ROUND(HIRE ROUND(HIRE
---------- ---------- ----------
1981/06/09 1981/06/01 1981/01/01
1981/11/17 1981/12/01 1982/01/01
1982/01/23 1982/02/01 1982/01/01
SQL> SELECT hiredate, TRUNC(hiredate,'MONTH'), TRUNC(hiredate,'YEAR')
       FROM emp
       WHERE deptno = 10 ;
HIREDATE   TRUNC(HIRE TRUNC(HIRE
---------- ---------- ----------
1981/06/09 1981/06/01 1981/01/01
1981/11/17 1981/11/01 1981/01/01
1982/01/23 1982/01/01 1982/01/01
SQL> ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MON-RR' ;
```

Quiz

1. EMP 테이블에서 사원의 이름, 입사일자, 부서번호를 출력하면서 입사일자가 포함된 주의 첫번째 날짜를 함께 표시 하시오. (주의 시작일자는 일요일로 한다.)

```
ENAME      HIREDATE      DEPTNO FIRST_DAY
---------- --------- ---------- ---------
SMITH      17-DEC-80         20 14-DEC-80
ALLEN      20-FEB-81         30 15-FEB-81
WARD       22-FEB-81         30 22-FEB-81
JONES      02-APR-81         20 29-MAR-81
MARTIN     28-SEP-81         30 27-SEP-81
BLAKE      01-MAY-81         30 26-APR-81
...
```

2. EMP 테이블의 ENAME 컬럼에서 대문자 S 가 들어간 횟수를 표시 하시오.

```
ENAME      Contains 'S'
---------- ------------
SMITH                 1
ALLEN                 0
WARD                  0
JONES                 1
MARTIN                0
BLAKE                 0
CLARK                 0
...
```

4. 변환 함수 및 조건부 표현식 사용

- *암시적 데이터 유형 변환 이해*

```
SQL> SELECT ename, sal, sal * '12'
       FROM emp
       WHERE deptno = 10 ;
ENAME             SAL    SAL*'12'
---------- ---------- ----------
CLARK            2450      29400
KING             5000      60000
MILLER           1300      15600
```

```
SQL> SELECT ename, sal, deptno
       FROM emp
       WHERE deptno = '10' ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
CLARK            2450         10
KING             5000         10
MILLER           1300         10
```

```
SQL> SET AUTOTRACE ON EXPLAIN
SQL> SELECT ename, hiredate, sal, deptno
       FROM emp
       WHERE hiredate LIKE '%82' ;
ENAME      HIREDATE         SAL     DEPTNO
---------- --------- ---------- ----------
SCOTT      09-DEC-82       3000         20
MILLER     23-JAN-82       1300         10
Execution Plan
----------------------------------------------------------
Plan hash value: 3956160932
--------------------------------------------------------------------------
| Id  | Operation         | Name | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------
|   0 | SELECT STATEMENT  |      |     1 |    21 |     3   (0)| 00:00:01 |
|*  1 |  TABLE ACCESS FULL| EMP  |     1 |    21 |     3   (0)| 00:00:01 |
--------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------
   1 - filter(INTERNAL_FUNCTION("HIREDATE") LIKE '%82')
SQL> SET AUTOTRACE OFF
```

- *명시적 데이터 유형 변환 : TO_CHAR 사용*

```
SQL> SELECT ename, hiredate, TO_CHAR(hiredate, 'YYYY/MM/DD') ,
                                TO_CHAR(hiredate, 'YYYY/MM/DD HH24:MI:SS')
       FROM emp
       WHERE deptno = 10 ;
ENAME      HIREDATE  TO_CHAR(HI TO_CHAR(HIREDATE,'Y
---------- --------- ---------- -------------------
CLARK      09-JUN-81 1981/06/09 1981/06/09 00:00:00
KING       17-NOV-81 1981/11/17 1981/11/17 00:00:00
MILLER     23-JAN-82 1982/01/23 1982/01/23 00:00:00
```

```
SQL> SELECT ename, hiredate, TO_CHAR(hiredate, 'DD Month YYYY', 'NLS_DATE_LANGUAGE=AMERICAN') ,
                             TO_CHAR(hiredate, 'fmDD Month YYYY', 'NLS_DATE_LANGUAGE=AMERICAN')
     FROM emp
     WHERE deptno = 10 ;
ENAME      HIREDATE  TO_CHAR(HIREDATE,'DDMONTHYYYY','NLS_DATE_LAN TO_CHAR(HIREDATE,'FMDDMONTHYYYY','NLS_DATE_L
---------- --------- -------------------------------------------- --------------------------------------------
CLARK      09-JUN-81 09 June      1981                            9 June 1981
KING       17-NOV-81 17 November  1981                            17 November 1981
MILLER     23-JAN-82 23 January   1982                            23 January 1982
SQL> SELECT ename, hiredate, TO_CHAR(hiredate, 'Q'),
                             TO_CHAR(hiredate,'D'),
                             TO_CHAR(hiredate,'W'),
                             TO_CHAR(hiredate,'Day')
     FROM emp
     WHERE deptno = 10 ;
ENAME      HIREDATE  T T T TO_CHAR(HIREDATE,'DAY')
---------- --------- - - - ------------------------------------
CLARK      09-JUN-81 2 3 2 Tuesday
KING       17-NOV-81 4 3 3 Tuesday
MILLER     23-JAN-82 1 7 4 Saturday
SQL> SELECT ename, sal, TO_CHAR(sal, '$99,999.00') ,
                        TO_CHAR(sal,'$00,000.00') ,
                        TO_CHAR(sal,'L99,999.00')
     FROM emp
     WHERE deptno = 10 ;
ENAME             SAL TO_CHAR(SAL TO_CHAR(SAL TO_CHAR(SAL,'L99,999
---------- ---------- ----------- ----------- --------------------
CLARK            2450   $2,450.00  $02,450.00            $2,450.00
KING             5000   $5,000.00  $05,000.00            $5,000.00
MILLER           1300   $1,300.00  $01,300.00            $1,300.00
```

- *명시적 데이터 유형 변환 : TO_NUMBER 사용*

```
SQL> VARIABLE b1 CHAR(2)
SQL> EXECUTE :b1 := '10'
SQL> SET AUTOTRACE ON EXPLAIN
SQL> SELECT empno, ename, deptno
       FROM emp
       WHERE deptno = :b1 ;
EMPNO ENAME          DEPTNO
---------- ---------- ----------
      7782 CLARK              10
      7839 KING               10
      7934 MILLER             10
Execution Plan
----------------------------------------------------------
Plan hash value: 3956160932
--------------------------------------------------------------------------
| Id  | Operation         | Name | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------
|   0 | SELECT STATEMENT  |      |     5 |    65 |     3   (0)| 00:00:01 |
|*  1 |  TABLE ACCESS FULL| EMP  |     5 |    65 |     3   (0)| 00:00:01 |
--------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------
   1 - filter("DEPTNO"=TO_NUMBER(:B1))
```

```
SQL> SELECT empno, ename, deptno
       FROM emp
       WHERE deptno = TO_NUMBER(:b1) ;
EMPNO ENAME          DEPTNO
---------- ---------- ----------
      7782 CLARK              10
      7839 KING               10
      7934 MILLER             10
Execution Plan
----------------------------------------------------------
Plan hash value: 3956160932
--------------------------------------------------------------------------
| Id  | Operation         | Name | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------
|   0 | SELECT STATEMENT  |      |     5 |    65 |     3   (0)| 00:00:01 |
|*  1 |  TABLE ACCESS FULL| EMP  |     5 |    65 |     3   (0)| 00:00:01 |
--------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------
   1 - filter("DEPTNO"=TO_NUMBER(:B1))
SQL> SET AUTOTRACE OFF
```

- *명시적 데이터 유형 변환 : TO_DATE 사용*

```
SQL> SELECT TRUNC('06-MAY-13','MONTH')
       FROM dual ;
SELECT TRUNC('06-MAY-13','MONTH')
             *
ERROR at line 1:
ORA-01722: invalid number
```

```
SQL> SELECT TRUNC(TO_DATE('06-MAY-13','DD-MON-RR'),'MONTH')
       FROM dual ;
TRUNC(TO_
---------
01-MAY-13
SQL> SELECT empno, ename, hiredate
       FROM emp
       WHERE hiredate = TO_DATE('01     May, 1981','DD Month, YYYY') ;
EMPNO ENAME      HIREDATE
---------- ---------- ---------
      7698 BLAKE      01-MAY-81
SQL> SELECT empno, ename, hiredate
       FROM emp
       WHERE hiredate = TO_DATE('01     May, 1981','fxDD Month, YYYY') ;
WHERE hiredate = TO_DATE('01     May, 1981','fxDD Month, YYYY')
                                *
ERROR at line 3:
ORA-01843: not a valid month
SQL> SELECT empno, ename, hiredate
       FROM emp
       WHERE hiredate = TO_DATE('01     May, 1981','fxDD     Month, YYYY') ;
EMPNO ENAME      HIREDATE
---------- ---------- ---------
      7698 BLAKE      01-MAY-81
```

- *일반 함수 사용*

```
SQL> SELECT ename, sal, comm, sal + NVL(comm,0)
       FROM emp
       WHERE deptno = 30 ;
ENAME             SAL       COMM SAL+NVL(COMM,0)
---------- ---------- ---------- ---------------
ALLEN            1600        300            1900
WARD             1250        500            1750
MARTIN           1250       1400            2650
BLAKE            2850 (NULL)                2850
TURNER           1500          0            1500
JAMES             950 (NULL)                 950
SQL> SELECT ename, mgr, NVL(mgr, 'No Manager')
       FROM emp WHERE deptno = 10 ;
SELECT ename, mgr, NVL(mgr, 'No Manager')
                             *
ERROR at line 1:
ORA-01722: invalid number
SQL> SELECT ename, mgr, NVL(TO_CHAR(mgr), 'No Manager')
       FROM emp WHERE deptno = 10 ;
ENAME             MGR NVL(TO_CHAR(MGR),'NOMANAGER')
---------- ---------- ----------------------------------------
CLARK            7839 7839
KING       (NULL)     No Manager
MILLER           7782 7782
```

```
SQL> SELECT ename, sal, comm, NVL2(comm, sal + comm, sal)
       FROM emp
       WHERE deptno = 30 ;
ENAME             SAL       COMM NVL2(COMM,SAL+COMM,SAL)
---------- ---------- ---------- -----------------------
ALLEN            1600        300                    1900
WARD             1250        500                    1750
MARTIN           1250       1400                    2650
BLAKE            2850 (NULL)                        2850
TURNER           1500          0                    1500
JAMES             950 (NULL)                         950
```

```
SQL> SELECT NULLIF ( 5, 5 ), NULLIF (5, 4)
       FROM dual ;
NULLIF(5,5) NULLIF(5,4)
----------- -----------
(NULL)                5
```

```
SQL> SELECT ename, comm, mgr, sal, NVL(comm,NVL(mgr,NVL(sal,-1)))
       FROM emp
       WHERE deptno = 10 ;
ENAME            COMM        MGR        SAL NVL(COMM,NVL(MGR,NVL(SAL,-1)))
---------- ---------- ---------- ---------- ------------------------------
CLARK      (NULL)           7839       2450                           7839
KING       (NULL)     (NULL)           5000                           5000
MILLER     (NULL)           7782       1300                           7782
```

```
SQL> SELECT ename, comm, mgr, sal, COALESCE (comm,mgr,sal,-1)
       FROM emp
       WHERE deptno = 10 ;
ENAME            COMM        MGR        SAL COALESCE(COMM,MGR,SAL,-1)
---------- ---------- ---------- ---------- -------------------------
CLARK      (NULL)           7839       2450                      7839
KING       (NULL)     (NULL)           5000                      5000
MILLER     (NULL)           7782       1300                      7782
```

- *조건부 표현식 사용*

```
SQL> SELECT ename, deptno, sal, CASE deptno WHEN 10 THEN sal * 1.1
                                            WHEN 20 THEN sal * 1.2
                                            WHEN 30 THEN sal * 1.3
                                    ELSE sal END AS inc_sal
       FROM emp ;
ENAME          DEPTNO        SAL    INC_SAL
---------- ---------- ---------- ----------
SMITH              20        800        960
ALLEN              30       1600       2080
WARD               30       1250       1625
JONES              20       2975       3570
MARTIN             30       1250       1625
BLAKE              30       2850       3705
CLARK              10       2450       2695
SCOTT              20       3000       3600
KING               10       5000       5500
TURNER             30       1500       1950
. . .
```

```
SQL> SELECT ename, deptno, sal, DECODE(deptno, 10, sal * 1.1 ,
                                                20, sal * 1.2 ,
                                                30, sal * 1.3 ,
                                         sal ) AS inc_sal
      FROM emp ;
ENAME          DEPTNO        SAL    INC_SAL
---------- ---------- ---------- ----------
SMITH              20        800        960
ALLEN              30       1600       2080
WARD               30       1250       1625
JONES              20       2975       3570
MARTIN             30       1250       1625
BLAKE              30       2850       3705
CLARK              10       2450       2695
SCOTT              20       3000       3600
KING               10       5000       5500
TURNER             30       1500       1950
...
```

```
SQL> SELECT ename, deptno, sal, CASE WHEN deptno IN (10,20) THEN sal * 1.1
                                      WHEN deptno = 30 THEN sal * 1.2 END AS inc_sal
      FROM emp ;
ENAME          DEPTNO        SAL    INC_SAL
---------- ---------- ---------- ----------
SMITH              20        800        880
ALLEN              30       1600       1920
WARD               30       1250       1500
JONES              20       2975     3272.5
MARTIN             30       1250       1500
BLAKE              30       2850       3420
CLARK              10       2450       2695
SCOTT              20       3000       3300
KING               10       5000       5500
TURNER             30       1500       1800
…
```

Quiz

1. EMP 테이블에서 사원의 이름, 입사일자, 및 근무 6개월 후 첫번째 월요일에 해당하는 급여 협상 날짜를 표시하시오. (협상 일자는 다음 결과와 같이 표시)

```
ENAME      HIREDATE  REVIEW
---------- --------- ---------------------------------------------------
SMITH      17-DEC-80 Monday, the Twenty-Second of June, 1981
ALLEN      20-FEB-81 Monday, the Twenty-Fourth of August, 1981
WARD       22-FEB-81 Monday, the Twenty-Fourth of August, 1981
JONES      02-APR-81 Monday, the Fifth of October, 1981
MARTIN     28-SEP-81 Monday, the Twenty-Ninth of March, 1982
...
```

2. EMP 테이블에서 사원의 이름 및 입사일자 및 입사일자의 요일을 표시 하시오. 단, 입사일의 요일을 기준으로 월요일부터 일요일 순으로 정렬된 결과를 표시 하시오.

```
ENAME      HIREDATE  DAY
---------- --------- ----------
MARTIN     28-SEP-81 Monday
CLARK      09-JUN-81 Tuesday
TURNER     08-SEP-81 Tuesday
KING       17-NOV-81 Tuesday
SMITH      17-DEC-80 Wednesday
ADAMS      12-JAN-83 Wednesday
JAMES      03-DEC-81 Thursday
JONES      02-APR-81 Thursday
FORD       03-DEC-81 Thursday
SCOTT      09-DEC-82 Thursday
ALLEN      20-FEB-81 Friday
BLAKE      01-MAY-81 Friday
MILLER     23-JAN-82 Saturday
WARD       22-FEB-81 Sunday
```

5. 그룹 함수를 사용하여 집계된 데이터 보고

- *그룹 함수 사용*

```
SQL> SELECT SUM(sal), AVG(sal), MAX(sal), MIN(sal), COUNT(sal)
        FROM emp ;
  SUM(SAL)   AVG(SAL)   MAX(SAL)   MIN(SAL) COUNT(SAL)
---------- ---------- ---------- ---------- ----------
     29025 2073.21429       5000        800         14
```

```
SQL> SELECT SUM(ename), AVG(ename), SUM(hiredate), AVG(hiredate)
        FROM emp ;
ERROR at line 1:
ORA-00932: inconsistent datatypes: expected NUMBER got DATE
```

```
SQL> SELECT MAX(ename), MIN(ename), MAX(hiredate), MIN(hiredate)
        FROM emp ;
MAX(ENAME) MIN(ENAME) MAX(HIRED MIN(HIRED
---------- ---------- --------- ---------
WARD       ADAMS      12-JAN-83 17-DEC-80
```

```
SQL> SELECT COUNT(*), COUNT(empno), COUNT(comm), COUNT(mgr)
        FROM emp ;
  COUNT(*) COUNT(EMPNO) COUNT(COMM) COUNT(MGR)
---------- ------------ ----------- ----------
        14           14           4         13
```

```
SQL> SELECT AVG(comm), SUM(comm) / 14
        FROM emp ;
 AVG(COMM) SUM(COMM)/14
---------- ------------
       550   157.142857
```

```
SQL> SELECT AVG(NVL(comm,0)) , SUM(comm) / 14
        FROM emp   ;
AVG(NVL(COMM,0)) SUM(COMM)/14
---------------- ------------
      157.142857   157.142857
```

```
SQL> SELECT COUNT(deptno), COUNT(DISTINCT deptno), SUM(deptno), SUM(DISTINCT deptno)
        FROM emp ;
COUNT(DEPTNO) COUNT(DISTINCTDEPTNO) SUM(DEPTNO) SUM(DISTINCTDEPTNO)
------------- --------------------- ----------- -------------------
           14                     3         310                  60
```

- *GROUP BY, HAVING 절 사용*

```
SQL> SELECT SUM(sal)
        FROM emp
        GROUP BY deptno ;
  SUM(SAL)
----------
      9400
     10875
      8750
```

```
SQL> SELECT deptno, SUM(sal)
       FROM emp
       GROUP BY deptno ;
    DEPTNO   SUM(SAL)
---------- ----------
        30       9400
        20      10875
        10       8750
```

```
SQL> SELECT deptno, job, SUM(sal)
       FROM emp
       GROUP BY deptno ;
ERROR at line 1:
ORA-00979: not a GROUP BY expression
```

```
SQL> SELECT deptno, job, SUM(sal)
       FROM emp
       GROUP BY deptno, job ;
    DEPTNO JOB         SUM(SAL)
---------- --------- ----------
        20 CLERK           1900
        30 SALESMAN        5600
        20 MANAGER         2975
        30 CLERK            950
        10 PRESIDENT       5000
        30 MANAGER         2850
        10 CLERK           1300
        10 MANAGER         2450
        20 ANALYST         6000
```

```
SQL> SELECT deptno, AVG(sal)
       FROM emp
       WHERE AVG(sal) > 1500
       GROUP BY deptno ;
      WHERE AVG(sal) > 1500
            *
ERROR at line 3:
ORA-00934: group function is not allowed here
```

```
SQL> SELECT deptno, AVG(sal)
       FROM emp
       HAVING AVG(sal) > 2500
       GROUP BY deptno ;
    DEPTNO   AVG(SAL)
---------- ----------
        10 2916.66667
```

• WHERE 절과 HAVING 절의 차이점 확인

```
SQL> SELECT deptno, SUM(sal)
       FROM emp
       WHERE deptno IN (20,30)
       GROUP BY deptno
       HAVING SUM(sal) > 10000 ;
    DEPTNO   SUM(SAL)
---------- ----------
        20      10875
SQL> SELECT deptno, SUM(sal)
       FROM emp
       GROUP BY deptno
       HAVING SUM(sal) > 10000
          AND deptno IN (20,30) ;
    DEPTNO   SUM(SAL)
---------- ----------
        20      10875

SQL> SELECT deptno, AVG(sal)
       FROM emp
       WHERE JOB = 'CLERK'
       GROUP BY deptno
       HAVING AVG(sal) > 1000 ;
    DEPTNO   AVG(SAL)
---------- ----------
        10       1300
SQL> SELECT deptno, AVG(sal)
       FROM emp
       GROUP BY deptno
       HAVING AVG(sal) > 1000
          AND JOB = 'CLERK' ;
ERROR at line 5:
ORA-00979: not a GROUP BY expression
```

• GROUP 함수의 중첩

```
SQL> SELECT SUM(sal)
       FROM emp
       GROUP BY deptno ;
  SUM(SAL)
----------
      9400
     10875
      8750
```

```
SQL> SELECT MAX(SUM(sal))
         FROM emp
         GROUP BY deptno ;
 MAX(SUM(SAL))
-------------
        10875
SQL> SELECT AVG(MAX(SUM(sal)))
         FROM emp
         GROUP BY deptno ;
ERROR at line 1:
ORA-00935: group function is nested too deeply
```

Quiz

1. EMP 테이블에서 부서별, 입사 연도별 급여의 합계를 다음과 같이 표시 하시오.

```
    DEPTNO       1980       1981       1982       1983      TOTAL
---------- ---------- ---------- ---------- ---------- ----------
        10          0       7450       1300          0       8750
        20        800       5975       3000       1100      10875
        30          0       9400          0          0       9400
```

6. 조인을 사용하여 여러 테이블의 데이터 표시

• Join 의 사용 (Equi Join)

```
SQL> SELECT empno, ename, deptno, dname, loc
       FROM emp, dept ;
SELECT empno, ename, deptno, dname, loc
                     *
ERROR at line 1:
ORA-00918: column ambiguously defined
```

```
SQL> SELECT emp.empno, emp.ename, emp.deptno, dept.deptno, dept.dname, dept.loc
       FROM emp, dept ;
     EMPNO ENAME          DEPTNO     DEPTNO DNAME          LOC
---------- ---------- ---------- ---------- -------------- -------------
      7369 SMITH              20         10 ACCOUNTING     NEW YORK
      7499 ALLEN              30         10 ACCOUNTING     NEW YORK
      7521 WARD               30         10 ACCOUNTING     NEW YORK
      7566 JONES              20         10 ACCOUNTING     NEW YORK
      7654 MARTIN             30         10 ACCOUNTING     NEW YORK
      7698 BLAKE              30         10 ACCOUNTING     NEW YORK
...
56 rows selected.
```

```
SQL> SELECT emp.empno, emp.ename, emp.deptno, dept.deptno, dept.dname, dept.loc
       FROM emp, dept
       ORDER BY 1 ;
     EMPNO ENAME          DEPTNO     DEPTNO DNAME          LOC
---------- ---------- ---------- ---------- -------------- -------------
      7369 SMITH              20         40 OPERATIONS     BOSTON
      7369 SMITH              20         30 SALES          CHICAGO
      7369 SMITH              20         20 RESEARCH       DALLAS
      7369 SMITH              20         10 ACCOUNTING     NEW YORK
      7499 ALLEN              30         30 SALES          CHICAGO
      7499 ALLEN              30         10 ACCOUNTING     NEW YORK
      7499 ALLEN              30         20 RESEARCH       DALLAS
      7499 ALLEN              30         40 OPERATIONS     BOSTON
...
56 rows selected.
```

```
SQL> SELECT emp.empno, emp.ename, emp.deptno, dept.deptno, dept.dname, dept.loc
       FROM emp, dept
       WHERE emp.deptno = dept.deptno ;
     EMPNO ENAME          DEPTNO     DEPTNO DNAME          LOC
---------- ---------- ---------- ---------- -------------- -------------
      7369 SMITH              20         20 RESEARCH       DALLAS
      7499 ALLEN              30         30 SALES          CHICAGO
      7521 WARD               30         30 SALES          CHICAGO
      7566 JONES              20         20 RESEARCH       DALLAS
      7654 MARTIN             30         30 SALES          CHICAGO
      7698 BLAKE              30         30 SALES          CHICAGO
      7782 CLARK              10         10 ACCOUNTING     NEW YORK
      7788 SCOTT              20         20 RESEARCH       DALLAS
      7839 KING               10         10 ACCOUNTING     NEW YORK
      7844 TURNER             30         30 SALES          CHICAGO
      7876 ADAMS              20         20 RESEARCH       DALLAS
      7900 JAMES              30         30 SALES          CHICAGO
      7902 FORD               20         20 RESEARCH       DALLAS
      7934 MILLER             10         10 ACCOUNTING     NEW YORK
```

- *ANSI Join 활용 (Equi Join)*

```
SQL> SELECT e.empno, e.ename, e.sal, d.deptno, d.dname, d.loc
       FROM emp e JOIN dept d
          ON e.deptno = d.deptno ;
     EMPNO ENAME             SAL     DEPTNO DNAME          LOC
---------- ---------- ---------- ---------- -------------- -------------
      7369 SMITH             800         20 RESEARCH       DALLAS
      7499 ALLEN            1600         30 SALES          CHICAGO
      7521 WARD             1250         30 SALES          CHICAGO
      7566 JONES            2975         20 RESEARCH       DALLAS
...
SQL> SELECT e.empno, e.ename, e.sal, deptno, d.dname, d.loc
       FROM emp e NATURAL JOIN dept d ;
     EMPNO ENAME             SAL     DEPTNO DNAME          LOC
---------- ---------- ---------- ---------- -------------- -------------
      7369 SMITH             800         20 RESEARCH       DALLAS
      7499 ALLEN            1600         30 SALES          CHICAGO
      7521 WARD             1250         30 SALES          CHICAGO
      7566 JONES            2975         20 RESEARCH       DALLAS
...
SQL> SELECT e.last_name, e.salary, d.department_id, d.department_name
       FROM employees e JOIN departments d
          ON e.department_id = d.department_id ;
LAST_NAME                     SALARY DEPARTMENT_ID DEPARTMENT_NAME
------------------------- ---------- ------------- ------------------------------
Whalen                          4400            10 Administration
Hartstein                      13000            20 Marketing
Fay                             6000            20 Marketing
Mourgos                         5800            50 Shipping
Vargas                          2500            50 Shipping
Matos                           2600            50 Shipping
Davies                          3100            50 Shipping
...
19 rows selected.
```

```
SQL> SELECT last_name, salary, department_id, department_name
       FROM employees e NATURAL JOIN departments d ;
LAST_NAME                     SALARY DEPARTMENT_ID DEPARTMENT_NAME
------------------------- ---------- ------------- ------------------------------
Kochhar                        17000            90 Executive
De Haan                        17000            90 Executive
Ernst                           6000            60 IT
Lorentz                         4200            60 IT
Rajs                            3500            50 Shipping
Davies                          3100            50 Shipping
Matos                           2600            50 Shipping
Vargas                          2500            50 Shipping
Abel                           11000            80 Sales
Taylor                          8600            80 Sales
Fay                             6000            20 Marketing
Gietz                           8300           110 Accounting
12 rows selected.
```

```
SQL> SELECT e.last_name, e.salary, d.department_id, d.department_name
        FROM employees e JOIN departments d
            ON e.department_id = d.department_id
          AND e.manager_id = d.manager_id ;
LAST_NAME                     SALARY DEPARTMENT_ID DEPARTMENT_NAME
------------------------- ---------- ------------- ------------------------------
Kochhar                        17000            90 Executive
De Haan                        17000            90 Executive
Ernst                           6000            60 IT
Lorentz                         4200            60 IT
Rajs                            3500            50 Shipping
Davies                          3100            50 Shipping
Matos                           2600            50 Shipping
Vargas                          2500            50 Shipping
Abel                           11000            80 Sales
Taylor                          8600            80 Sales
Fay                             6000            20 Marketing
Gietz                           8300           110 Accounting
12 rows selected.
```

```
SQL> SELECT last_name, salary, department_id, department_name
        FROM employees e JOIN departments d
          USING (department_id)   ;
LAST_NAME                     SALARY DEPARTMENT_ID DEPARTMENT_NAME
------------------------- ---------- ------------- ------------------------------
Whalen                          4400            10 Administration
Hartstein                      13000            20 Marketing
Fay                             6000            20 Marketing
Mourgos                         5800            50 Shipping
Vargas                          2500            50 Shipping
Matos                           2600            50 Shipping
Davies                          3100            50 Shipping
Rajs                            3500            50 Shipping
...
```

• NATURAL JOIN, USING 절 사용 시 주의 사항

```
SQL> SELECT e.empno, e.ename, e.sal, d.deptno, d.dname, d.loc
        FROM emp e NATURAL JOIN dept d ;
SELECT e.empno, e.ename, e.sal, d.deptno, d.dname, d.loc
                                *
ERROR at line 1:
ORA-25155: column used in NATURAL join cannot have qualifier
```

```
SQL> SELECT e.last_name, e.salary, d.department_id, d.department_name
        FROM employees e JOIN departments d
          USING (department_id)   ;
SELECT e.last_name, e.salary, d.department_id, d.department_name
                              *
ERROR at line 1:
ORA-25154: column part of USING clause cannot have qualifier
```

• Non-Equi Join 활용

```
SQL> SELECT e.ename, e.sal, s.grade
       FROM emp e JOIN salgrade s
          ON e.sal BETWEEN s.losal AND s.hisal ;
ENAME             SAL      GRADE
---------- ---------- ----------
SMITH             800          1
JAMES             950          1
ADAMS            1100          1
WARD             1250          2
MARTIN           1250          2
MILLER           1300          2
...
SQL> SELECT e.ename, e.sal, s.grade
       FROM emp e, salgrade s
       WHERE e.sal BETWEEN s.losal AND s.hisal ;
ENAME             SAL      GRADE
---------- ---------- ----------
SMITH             800          1
JAMES             950          1
ADAMS            1100          1
WARD             1250          2
MARTIN           1250          2
MILLER           1300          2
...
```

• Self Join 활용

- 'JONES' 사원보다 많은 급여를 받는 사원을 표시

```
SQL> SELECT sal
       FROM emp j
       WHERE ename = 'JONES' ;
      SAL
----------
      2975
SQL> SELECT ename, sal, deptno
       FROM emp e
       WHERE sal > 2975 ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
SCOTT            3000         20
KING             5000         10
FORD             3000         20
SQL> SELECT e.ename, e.sal, e.deptno
       FROM emp e, emp j
       WHERE j.ename = 'JONES'
         AND e.sal > j.sal ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
SCOTT            3000         20
KING             5000         10
FORD             3000         20
```

```
SQL> SELECT e.ename, e.sal, e.deptno
       FROM emp e JOIN emp j
          ON j.ename = 'JONES'
         AND e.sal > j.sal ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
SCOTT            3000         20
KING             5000         10
FORD             3000         20
```

- 각 사원들의 이름과 해당 사원의 상사 이름을 함께 표시

```
SQL> SELECT e.ename || ' works for ' ||m.ename
       FROM emp e JOIN emp m
          ON e.mgr = m.empno ;
E.ENAME||'WORKSFOR'||M.ENAME
-------------------------------
FORD works for JONES
SCOTT works for JONES
JAMES works for BLAKE
TURNER works for BLAKE
MARTIN works for BLAKE
WARD works for BLAKE
ALLEN works for BLAKE
MILLER works for CLARK
ADAMS works for SCOTT
CLARK works for KING
BLAKE works for KING
JONES works for KING
SMITH works for FORD
```

• *Outer Join 활용*

```
SQL> SELECT deptno, dname, loc FROM dept ;
    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON
```

```
SQL> SELECT DISTINCT deptno FROM emp ;
    DEPTNO
----------
        30
        20
        10
```

```
SQL> SELECT e.empno, e.ename, e.sal, e.deptno, d.deptno, d.dname, d.loc
       FROM emp e JOIN dept d
          ON e.deptno = d.deptno ;
EMPNO ENAME             SAL     DEPTNO     DEPTNO DNAME          LOC
---------- ---------- ---------- ---------- ---------- -------------- -------------
      7369 SMITH             800         20         20 RESEARCH       DALLAS
      7499 ALLEN            1600         30         30 SALES          CHICAGO
      7521 WARD             1250         30         30 SALES          CHICAGO
      7566 JONES            2975         20         20 RESEARCH       DALLAS
      7654 MARTIN           1250         30         30 SALES          CHICAGO
      7698 BLAKE            2850         30         30 SALES          CHICAGO
      7782 CLARK            2450         10         10 ACCOUNTING     NEW YORK
      7788 SCOTT            3000         20         20 RESEARCH       DALLAS
      7839 KING             5000         10         10 ACCOUNTING     NEW YORK
      7844 TURNER           1500         30         30 SALES          CHICAGO
      7876 ADAMS            1100         20         20 RESEARCH       DALLAS
      7900 JAMES             950         30         30 SALES          CHICAGO
      7902 FORD             3000         20         20 RESEARCH       DALLAS
      7934 MILLER           1300         10         10 ACCOUNTING     NEW YORK
14 rows selected.
SQL> SELECT e.empno, e.ename, e.sal, e.deptno, d.deptno, d.dname, d.loc
       FROM emp e INNER JOIN dept d
          ON e.deptno = d.deptno ;
EMPNO ENAME             SAL     DEPTNO     DEPTNO DNAME          LOC
---------- ---------- ---------- ---------- ---------- -------------- -------------
      7369 SMITH             800         20         20 RESEARCH       DALLAS
      7499 ALLEN            1600         30         30 SALES          CHICAGO
      7521 WARD             1250         30         30 SALES          CHICAGO
      7566 JONES            2975         20         20 RESEARCH       DALLAS
      7654 MARTIN           1250         30         30 SALES          CHICAGO
      7698 BLAKE            2850         30         30 SALES          CHICAGO
      7782 CLARK            2450         10         10 ACCOUNTING     NEW YORK
      7788 SCOTT            3000         20         20 RESEARCH       DALLAS
      7839 KING             5000         10         10 ACCOUNTING     NEW YORK
      7844 TURNER           1500         30         30 SALES          CHICAGO
      7876 ADAMS            1100         20         20 RESEARCH       DALLAS
      7900 JAMES             950         30         30 SALES          CHICAGO
      7902 FORD             3000         20         20 RESEARCH       DALLAS
      7934 MILLER           1300         10         10 ACCOUNTING     NEW YORK
14 rows selected.

SQL> SELECT e.empno, e.ename, e.sal, e.deptno, d.deptno, d.dname, d.loc
       FROM emp e RIGHT OUTER JOIN dept d
          ON e.deptno = d.deptno ;
EMPNO ENAME             SAL     DEPTNO     DEPTNO DNAME          LOC
---------- ---------- ---------- ---------- ---------- -------------- -------------
      7369 SMITH             800         20         20 RESEARCH       DALLAS
      7499 ALLEN            1600         30         30 SALES          CHICAGO
      7521 WARD             1250         30         30 SALES          CHICAGO
      7566 JONES            2975         20         20 RESEARCH       DALLAS
      7654 MARTIN           1250         30         30 SALES          CHICAGO
      7698 BLAKE            2850         30         30 SALES          CHICAGO
      7782 CLARK            2450         10         10 ACCOUNTING     NEW YORK
      7788 SCOTT            3000         20         20 RESEARCH       DALLAS
...
(NULL)     (NULL)     (NULL)     (NULL)             40 OPERATIONS     BOSTON
15 rows selected.
```

```
SQL> SELECT department_id, department_name
       FROM departments ;
DEPARTMENT_ID DEPARTMENT_NAME
------------- ------------------------------
           10 Administration
           20 Marketing
           50 Shipping
           60 IT
           80 Sales
           90 Executive
          110 Accounting
          190 Contracting
8 rows selected.
SQL> SELECT DISTINCT department_id FROM employees ;
DEPARTMENT_ID
-------------
(NULL)
           90
           20
          110
           50
           80
           60
           10
SQL> SELECT e.employee_id, e.last_name, e.salary, e.department_id, d.department_id, d.department_name
       FROM employees e RIGHT OUTER JOIN departments d
          ON e.department_id = d.department_id ;
EMPLOYEE_ID LAST_NAME                     SALARY DEPARTMENT_ID DEPARTMENT_ID DEPARTMENT_NAME
----------- ------------------------- ---------- ------------- ------------- ------------------------------
        200 Whalen                          4400            10            10 Administration
        201 Hartstein                      13000            20            20 Marketing
        202 Fay                             6000            20            20 Marketing
        124 Mourgos                         5800            50            50 Shipping
        144 Vargas                          2500            50            50 Shipping
        143 Matos                           2600            50            50 Shipping
        142 Davies                          3100            50            50 Shipping
        141 Rajs                            3500            50            50 Shipping
...
(NULL)      (NULL)                    (NULL)     (NULL)                  190 Contracting
20 rows selected.
SQL> SELECT e.employee_id, e.last_name, e.salary, e.department_id, d.department_id, d.department_name
       FROM employees e LEFT OUTER JOIN departments d
          ON e.department_id = d.department_id ;
EMPLOYEE_ID LAST_NAME                     SALARY DEPARTMENT_ID DEPARTMENT_ID DEPARTMENT_NAME
----------- ------------------------- ---------- ------------- ------------- ------------------------------
        200 Whalen                          4400            10            10 Administration
        202 Fay                             6000            20            20 Marketing
        201 Hartstein                      13000            20            20 Marketing
        144 Vargas                          2500            50            50 Shipping
        143 Matos                           2600            50            50 Shipping
        142 Davies                          3100            50            50 Shipping
        141 Rajs                            3500            50            50 Shipping
        124 Mourgos                         5800            50            50 Shipping
        107 Lorentz                         4200            60            60 IT
        104 Ernst                           6000            60            60 IT
...
        178 Grant                           7000 (NULL)        (NULL)        (NULL)
20 rows selected.
```

```
SQL> SELECT e.employee_id, e.last_name, e.salary, e.department_id, d.department_id, d.department_name
       FROM employees e FULL OUTER JOIN departments d
          ON e.department_id = d.department_id ;
EMPLOYEE_ID LAST_NAME                     SALARY DEPARTMENT_ID DEPARTMENT_ID DEPARTMENT_NAME
----------- ------------------------- ---------- ------------- ------------- ------------------------------
        100 King                           24000            90            90 Executive
        101 Kochhar                        17000            90            90 Executive
        102 De Haan                        17000            90            90 Executive
        103 Hunold                          9000            60            60 IT
        104 Ernst                           6000            60            60 IT
        107 Lorentz                         4200            60            60 IT
        124 Mourgos                         5800            50            50 Shipping
        141 Rajs                            3500            50            50 Shipping
        142 Davies                          3100            50            50 Shipping
        143 Matos                           2600            50            50 Shipping
        144 Vargas                          2500            50            50 Shipping
        149 Zlotkey                        10500            80            80 Sales
        174 Abel                           11000            80            80 Sales
        176 Taylor                          8600            80            80 Sales
        178 Grant                           7000 (NULL)        (NULL)        (NULL)
        200 Whalen                          4400            10            10 Administration
        201 Hartstein                      13000            20            20 Marketing
        202 Fay                             6000            20            20 Marketing
        205 Higgins                        12000           110           110 Accounting
        206 Gietz                           8300           110           110 Accounting
(NULL)      (NULL)                    (NULL)     (NULL)                  190 Contracting
21 rows selected.
```

```
SQL> SELECT e.employee_id, e.last_name, e.salary, e.department_id, d.department_id, d.department_name
       FROM employees e , departments d
       WHERE e.department_id (+) = d.department_id ;
EMPLOYEE_ID LAST_NAME                     SALARY DEPARTMENT_ID DEPARTMENT_ID DEPARTMENT_NAME
----------- ------------------------- ---------- ------------- ------------- ------------------------------
        200 Whalen                          4400            10            10 Administration
        201 Hartstein                      13000            20            20 Marketing
        202 Fay                             6000            20            20 Marketing
        124 Mourgos                         5800            50            50 Shipping
        144 Vargas                          2500            50            50 Shipping
        143 Matos                           2600            50            50 Shipping
        142 Davies                          3100            50            50 Shipping
...
(NULL)      (NULL)                    (NULL)     (NULL)                  190 Contracting
```

```
SQL> SELECT e.employee_id, e.last_name, e.salary, e.department_id, d.department_id, d.department_name
       FROM employees e , departments d
       WHERE e.department_id = d.department_id (+) ;
EMPLOYEE_ID LAST_NAME                     SALARY DEPARTMENT_ID DEPARTMENT_ID DEPARTMENT_NAME
----------- ------------------------- ---------- ------------- ------------- ------------------------------
        200 Whalen                          4400            10            10 Administration
        202 Fay                             6000            20            20 Marketing
        201 Hartstein                      13000            20            20 Marketing
        144 Vargas                          2500            50            50 Shipping
        143 Matos                           2600            50            50 Shipping
        142 Davies                          3100            50            50 Shipping
...
        178 Grant                           7000 (NULL)        (NULL)        (NULL)
20 rows selected.
```

```
SQL> SELECT e.employee_id, e.last_name, e.salary, e.department_id, d.department_id, d.department_name
        FROM employees e , departments d
        WHERE e.department_id (+) = d.department_id (+) ;
ERROR at line 3:
ORA-01468: a predicate may reference only one outer-joined table
```

- *Outer Join 사용 시 주의 사항 확인*

```
SQL> SELECT e.employee_id, e.last_name, e.salary, e.department_id, d.department_id, d.department_name
        FROM employees e , departments d
        WHERE e.department_id (+) = d.department_id
          AND e.salary > 10000 ;
EMPLOYEE_ID LAST_NAME                     SALARY DEPARTMENT_ID DEPARTMENT_ID DEPARTMENT_NAME
----------- ------------------------- ---------- ------------- ------------- ------------------------------
        201 Hartstein                      13000            20            20 Marketing
        149 Zlotkey                        10500            80            80 Sales
        174 Abel                           11000            80            80 Sales
        100 King                           24000            90            90 Executive
        102 De Haan                        17000            90            90 Executive
        101 Kochhar                        17000            90            90 Executive
        205 Higgins                        12000           110           110 Accounting
7 rows selected.
```

```
SQL> SELECT e.employee_id, e.last_name, e.salary, e.department_id, d.department_id, d.department_name
        FROM employees e , departments d
        WHERE e.department_id (+) = d.department_id
          AND ( e.salary > 10000 OR e.salary IS NULL ) ;
EMPLOYEE_ID LAST_NAME                     SALARY DEPARTMENT_ID DEPARTMENT_ID DEPARTMENT_NAME
----------- ------------------------- ---------- ------------- ------------- ------------------------------
        201 Hartstein                      13000            20            20 Marketing
        174 Abel                           11000            80            80 Sales
        149 Zlotkey                        10500            80            80 Sales
        102 De Haan                        17000            90            90 Executive
        100 King                           24000            90            90 Executive
        101 Kochhar                        17000            90            90 Executive
        205 Higgins                        12000           110           110 Accounting
(NULL)      (NULL)                    (NULL)     (NULL)                  190 Contracting
```

```
SQL> SELECT e.employee_id, e.last_name, e.salary, e.department_id, d.department_id, d.department_name
        FROM employees e , departments d
        WHERE e.department_id (+) = d.department_id
          AND e.salary (+) > 10000 ;
EMPLOYEE_ID LAST_NAME                     SALARY DEPARTMENT_ID DEPARTMENT_ID DEPARTMENT_NAME
----------- ------------------------- ---------- ------------- ------------- ------------------------------
        201 Hartstein                      13000            20            20 Marketing
        174 Abel                           11000            80            80 Sales
        149 Zlotkey                        10500            80            80 Sales
        102 De Haan                        17000            90            90 Executive
        100 King                           24000            90            90 Executive
        101 Kochhar                        17000            90            90 Executive
        205 Higgins                        12000           110           110 Accounting
(NULL)      (NULL)                    (NULL)     (NULL)                  190 Contracting
```

```
SQL> SELECT e.employee_id, e.last_name, e.salary, e.department_id, d.department_id, d.department_name
       FROM employees e RIGHT OUTER JOIN departments d
          ON e.department_id = d.department_id
       WHERE e.salary > 10000 ;
```

```
EMPLOYEE_ID LAST_NAME                     SALARY DEPARTMENT_ID DEPARTMENT_ID DEPARTMENT_NAME
----------- ------------------------- ---------- ------------- ------------- ------------------------------
        201 Hartstein                      13000            20            20 Marketing
        149 Zlotkey                        10500            80            80 Sales
        174 Abel                           11000            80            80 Sales
        100 King                           24000            90            90 Executive
        102 De Haan                        17000            90            90 Executive
        101 Kochhar                        17000            90            90 Executive
        205 Higgins                        12000           110           110 Accounting
7 rows selected.
```

```
SQL> SELECT e.employee_id, e.last_name, e.salary, e.department_id, d.department_id, d.department_name
       FROM employees e RIGHT OUTER JOIN departments d
          ON e.department_id = d.department_id
         AND e.salary > 10000 ;
```

```
EMPLOYEE_ID LAST_NAME                     SALARY DEPARTMENT_ID DEPARTMENT_ID DEPARTMENT_NAME
----------- ------------------------- ---------- ------------- ------------- ------------------------------
        201 Hartstein                      13000            20            20 Marketing
        174 Abel                           11000            80            80 Sales
        149 Zlotkey                        10500            80            80 Sales
        102 De Haan                        17000            90            90 Executive
        100 King                           24000            90            90 Executive
        101 Kochhar                        17000            90            90 Executive
        205 Higgins                        12000           110           110 Accounting
(NULL)      (NULL)                    (NULL)     (NULL)                  190 Contracting
```

- *Cross Join 활용*

```
SQL> SELECT e.empno, e.ename, e.deptno, d.deptno, d.dname
       FROM emp e CROSS JOIN dept d ;
```

```
    EMPNO ENAME          DEPTNO     DEPTNO DNAME
---------- ---------- ---------- ---------- --------------
      7369 SMITH              20         10 ACCOUNTING
      7499 ALLEN              30         10 ACCOUNTING
      7521 WARD               30         10 ACCOUNTING
      7566 JONES              20         10 ACCOUNTING
      7654 MARTIN             30         10 ACCOUNTING
      7698 BLAKE              30         10 ACCOUNTING
      7782 CLARK              10         10 ACCOUNTING
...
56 rows selected.
```

```
SQL> SELECT e.empno, e.ename, e.deptno, d.deptno, d.dname
       FROM emp e , dept d ;
```

※ Cross Join 과 동일한 결과

7. Subquery 를 사용하여 Query 해결

• *Single Row Subquery 사용*

- JONES 사원보다 더 많은 급여를 받는 사원들 검색

```
SQL> SELECT sal
       FROM emp
       WHERE ename = 'JONES' ;
       SAL
----------
      2975
SQL> SELECT ename, sal, deptno
       FROM emp
       WHERE sal > 2975 ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
SCOTT            3000         20
KING             5000         10
FORD             3000         20
SQL> SELECT ename, sal, deptno
       FROM emp
       WHERE sal > ( SELECT sal
                       FROM emp
                       WHERE ename = 'JONES' ) ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
SCOTT            3000         20
KING             5000         10
FORD             3000         20
```

- EMP 테이블에서 평균 급여보다 더 많은 급여를 받는 사원들 검색

```
SQL> SELECT ename, sal, deptno
       FROM emp
       WHERE sal > ( SELECT AVG(sal) FROM emp ) ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
JONES            2975         20
BLAKE            2850         30
CLARK            2450         10
SCOTT            3000         20
KING             5000         10
FORD             3000         20
```

- EMP 테이블에서 부서별 최소 급여와 동일한 급여를 갖는 사원들 검색

```
SQL> SELECT ename, sal, deptno
       FROM emp
       WHERE sal = ( SELECT MIN(sal)
                       FROM emp
                       GROUP BY deptno ) ;
ERROR at line 3:
ORA-01427: single-row subquery returns more than one row
```

- *Multiple Rows Subquery 사용*

```
SQL> SELECT ename, sal, deptno
       FROM emp
       WHERE sal IN ( SELECT MIN(sal)
                      FROM emp
                      GROUP BY deptno ) ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
SMITH             800         20
JAMES             950         30
MILLER           1300         10
SQL> SELECT ename, sal, deptno
       FROM emp
       WHERE sal IN ( 950, 800, 1300 ) ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
SMITH             800         20
JAMES             950         30
MILLER           1300         10
SQL> SELECT ename, sal, deptno
       FROM emp
       WHERE sal = 950
           OR sal = 800
           OR sal = 1300 ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
SMITH             800         20
JAMES             950         30
MILLER           1300         10
```

- 임의의 부서별 평균 급여보다 더 많은 급여를 받는 사원들 검색

```
SQL> SELECT ename, sal, deptno
       FROM emp
       WHERE sal >ANY ( SELECT AVG(sal)
                          FROM emp
                          GROUP BY deptno ) ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
KING             5000         10
FORD             3000         20
SCOTT            3000         20
JONES            2975         20
BLAKE            2850         30
CLARK            2450         10
ALLEN            1600         30
```

```
SQL> SELECT ename, sal, deptno
       FROM emp
       WHERE sal >ANY ( 1566, 2175, 2916 ) ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
ALLEN            1600         30
JONES            2975         20
BLAKE            2850         30
CLARK            2450         10
SCOTT            3000         20
KING             5000         10
FORD             3000         20
```

```
SQL> SELECT ename, sal, deptno
       FROM emp
       WHERE sal > 1566
            OR sal > 2175
            OR sal > 2916 ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
ALLEN            1600         30
JONES            2975         20
BLAKE            2850         30
CLARK            2450         10
SCOTT            3000         20
KING             5000         10
FORD             3000         20
```

- 각각의 모든 부서별 평균 급여보다 더 많은 급여를 받는 사원들 검색

```
SQL> SELECT ename, sal, deptno
       FROM emp
       WHERE sal >ALL ( SELECT AVG(sal)
                        FROM emp
                        GROUP BY deptno ) ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
JONES            2975         20
SCOTT            3000         20
FORD             3000         20
KING             5000         10
```

```
SQL> SELECT ename, sal, deptno
       FROM emp
       WHERE sal >ALL ( 1566, 2175, 2916 ) ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
JONES            2975         20
SCOTT            3000         20
KING             5000         10
FORD             3000         20
```

```
SQL> SELECT ename, sal, deptno
       FROM emp
       WHERE sal > 1566
         AND sal > 2175
         AND sal > 2916 ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
JONES            2975         20
SCOTT            3000         20
KING             5000         10
FORD             3000         20
```

```
SQL> SELECT ename, sal, deptno
       FROM emp
       WHERE sal > ( SELECT MIN(AVG(sal))
                     FROM emp
                     GROUP BY deptno ) ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
ALLEN            1600         30
JONES            2975         20
BLAKE            2850         30
CLARK            2450         10
SCOTT            3000         20
KING             5000         10
FORD             3000         20
```

```
SQL> SELECT ename, sal, deptno
       FROM emp
       WHERE sal > ( SELECT MAX(AVG(sal))
                     FROM emp
                     GROUP BY deptno ) ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
JONES            2975         20
SCOTT            3000         20
KING             5000         10
FORD             3000         20
```

- *Subquery 와 NOT IN 연산 사용 시 주의 사항*

```
SQL> SELECT ename, mgr, sal, deptno
       FROM emp
       WHERE empno IN ( SELECT mgr FROM emp ) ;
ENAME             MGR        SAL     DEPTNO
---------- ---------- ---------- ----------
FORD             7566       3000         20
BLAKE            7839       2850         30
KING       (NULL)           5000         10
JONES            7839       2975         20
SCOTT            7566       3000         20
CLARK            7839       2450         10
```

```
SQL> SELECT ename, mgr, sal, deptno
       FROM emp
       WHERE empno NOT IN ( SELECT mgr FROM emp ) ;
no rows selected
SQL> SELECT ename, sal, deptno
       FROM emp
       WHERE deptno IN (10,20,NULL) ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
SMITH             800         20
JONES            2975         20
CLARK            2450         10
SCOTT            3000         20
KING             5000         10
ADAMS            1100         20
FORD             3000         20
MILLER           1300         10

SQL> SELECT ename, sal, deptno
       FROM emp
       WHERE deptno = 10
            OR deptno = 20
            OR deptno = NULL ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
SMITH             800         20
JONES            2975         20
CLARK            2450         10
SCOTT            3000         20
KING             5000         10
ADAMS            1100         20
FORD             3000         20
MILLER           1300         10

SQL> SELECT ename, sal, deptno
       FROM emp
       WHERE deptno NOT IN (10,20,NULL) ;
no rows selected

SQL> SELECT ename, sal, deptno
       FROM emp
       WHERE deptno != 10
           AND deptno != 20
           AND deptno != NULL ;

no rows selected
```

```
SQL> SELECT ename, mgr, sal, deptno
      FROM emp
      WHERE empno NOT IN ( SELECT NVL(mgr,-1) FROM emp ) ;
ENAME             MGR        SAL     DEPTNO
---------- ---------- ---------- ----------
TURNER           7698       1500         30
WARD             7698       1250         30
MARTIN           7698       1250         30
ALLEN            7698       1600         30
MILLER           7782       1300         10
SMITH            7902        800         20
ADAMS            7788       1100         20
JAMES            7698        950         30
```

Quiz

1. EMP 테이블에서 소속 부서의 평균 급여보다 더 많은 급여를 받는 사원을 표시 하시오.

```
     EMPNO ENAME             SAL     DEPTNO
---------- ---------- ---------- ----------
      7499 ALLEN            1600         30
      7566 JONES            2975         20
      7698 BLAKE            2850         30
      7788 SCOTT            3000         20
      7839 KING             5000         10
      7902 FORD             3000         20
6 rows selected.
```

2. EMP 테이블에서 부서번호 10번 사원들의 정보를 EMPNO 컬럼을 기준으로 정렬하여 다음과 같이 표시하시오. (사원들의 정보를 표시 하면서 각각의 급여를 누적하여 TOTAL 컬럼을 생성)

```
     EMPNO ENAME          DEPTNO        SAL      TOTAL
---------- ---------- ---------- ---------- ----------
      7782 CLARK              10       2450       2450
      7839 KING               10       5000       7450
      7934 MILLER             10       1300       8750
```

8. 집합 연산자 사용

• UNION 사용

```
SQL> SELECT deptno, ename, sal
       FROM emp
       WHERE deptno IN (10,30)
       UNION
       SELECT deptno, ename, sal
       FROM emp
       WHERE deptno IN (20,30) ;
    DEPTNO ENAME             SAL
---------- ---------- ----------
        10 CLARK            2450
        10 KING             5000
        10 MILLER           1300
        20 ADAMS            1100
        20 FORD             3000
        20 JONES            2975
        20 SCOTT            3000
        20 SMITH             800
        30 ALLEN            1600
        30 BLAKE            2850
        30 JAMES             950
        30 MARTIN           1250
        30 TURNER           1500
        30 WARD             1250
14 rows selected.
SQL> SELECT sal, ename, deptno
       FROM emp
       WHERE deptno IN (10,30)
       UNION
       SELECT sal, ename, deptno
       FROM emp
       WHERE deptno IN (20,30) ;
       SAL ENAME          DEPTNO
---------- ---------- ----------
       800 SMITH              20
       950 JAMES              30
      1100 ADAMS              20
      1250 MARTIN             30
      1250 WARD               30
      1300 MILLER             10
      1500 TURNER             30
      1600 ALLEN              30
      2450 CLARK              10
      2850 BLAKE              30
      2975 JONES              20
      3000 FORD               20
      3000 SCOTT              20
      5000 KING               10
14 rows selected.
```

- *UNION ALL 사용*

```
SQL> SELECT deptno, ename, sal
       FROM emp
       WHERE deptno IN (10,30)
       UNION ALL
       SELECT deptno, ename, sal
       FROM emp
       WHERE deptno IN (20,30) ;
    DEPTNO ENAME             SAL
---------- ---------- ----------
        30 ALLEN            1600
        30 WARD             1250
        30 MARTIN           1250
        30 BLAKE            2850
        10 CLARK            2450
        10 KING             5000
        30 TURNER           1500
        30 JAMES             950
        10 MILLER           1300
        20 SMITH             800
        30 ALLEN            1600
        30 WARD             1250
        20 JONES            2975
        30 MARTIN           1250
        30 BLAKE            2850
        20 SCOTT            3000
        30 TURNER           1500
        20 ADAMS            1100
        30 JAMES             950
        20 FORD             3000
20 rows selected.
```

- *INTERSECT 사용*

```
SQL> SELECT ename, sal, deptno
       FROM emp
       WHERE deptno IN (10,30)
       INTERSECT
       SELECT ename, sal, deptno
       FROM emp
       WHERE deptno IN (20,30) ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
ALLEN            1600         30
BLAKE            2850         30
JAMES             950         30
MARTIN           1250         30
TURNER           1500         30
WARD             1250         30
```

- *MINUS 사용*

```
SQL> SELECT ename, sal, deptno
       FROM emp
       WHERE deptno IN (10,30)
       MINUS
       SELECT ename, sal, deptno
       FROM emp
       WHERE deptno IN (20,30) ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
CLARK            2450         10
KING             5000         10
MILLER           1300         10
```

- *집합 연산자의 중복행 제거 확인*

```
SQL> SET AUTOTRACE TRACEONLY EXPLAIN
SQL> SELECT deptno, ename, sal
       FROM emp
       WHERE deptno IN (10,30)
       UNION ALL
       SELECT deptno, ename, sal
       FROM emp
       WHERE deptno IN (20,30) ;
--------------------------------------------------------------------------
| Id  | Operation          | Name | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |      |    20 |   260 |     6  (50)| 00:00:01 |
|   1 |  UNION-ALL         |      |       |       |            |          |
|*  2 |   TABLE ACCESS FULL| EMP  |     9 |   117 |     3   (0)| 00:00:01 |
|*  3 |   TABLE ACCESS FULL| EMP  |    11 |   143 |     3   (0)| 00:00:01 |
--------------------------------------------------------------------------
SQL> SELECT deptno, ename, sal
       FROM emp
       WHERE deptno IN (10,30)
       UNION
       SELECT deptno, ename, sal
       FROM emp
       WHERE deptno IN (20,30) ;
---------------------------------------------------------------------------
| Id  | Operation           | Name | Rows  | Bytes | Cost (%CPU)| Time     |
---------------------------------------------------------------------------
|   0 | SELECT STATEMENT    |      |    20 |   260 |     8  (63)| 00:00:01 |
|   1 |  SORT UNIQUE        |      |    20 |   260 |     8  (63)| 00:00:01 |
|   2 |   UNION-ALL         |      |       |       |            |          |
|*  3 |    TABLE ACCESS FULL| EMP  |     9 |   117 |     3   (0)| 00:00:01 |
|*  4 |    TABLE ACCESS FULL| EMP  |    11 |   143 |     3   (0)| 00:00:01 |
---------------------------------------------------------------------------
```

```
SQL> SELECT deptno, ename, sal
       FROM emp
       WHERE deptno IN (10,30)
       INTERSECT
       SELECT deptno, ename, sal
       FROM emp
       WHERE deptno IN (20,30) ;
-----------------------------------------------------------------------------
| Id  | Operation            | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------
|   0 | SELECT STATEMENT     |      |     9 |   260 |     8  (63)| 00:00:01 |
|   1 |  INTERSECTION        |      |       |       |            |          |
|   2 |   SORT UNIQUE        |      |     9 |   117 |     4  (25)| 00:00:01 |
|*  3 |    TABLE ACCESS FULL| EMP  |     9 |   117 |     3   (0)| 00:00:01 |
|   4 |   SORT UNIQUE        |      |    11 |   143 |     4  (25)| 00:00:01 |
|*  5 |    TABLE ACCESS FULL| EMP  |    11 |   143 |     3   (0)| 00:00:01 |
-----------------------------------------------------------------------------
SQL> SELECT deptno, ename, sal
       FROM emp
       WHERE deptno IN (10,30)
       MINUS
       SELECT deptno, ename, sal
       FROM emp
       WHERE deptno IN (20,30) ;
-----------------------------------------------------------------------------
| Id  | Operation            | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------
|   0 | SELECT STATEMENT     |      |     9 |   260 |     8  (63)| 00:00:01 |
|   1 |  MINUS               |      |       |       |            |          |
|   2 |   SORT UNIQUE        |      |     9 |   117 |     4  (25)| 00:00:01 |
|*  3 |    TABLE ACCESS FULL| EMP  |     9 |   117 |     3   (0)| 00:00:01 |
|   4 |   SORT UNIQUE        |      |    11 |   143 |     4  (25)| 00:00:01 |
|*  5 |    TABLE ACCESS FULL| EMP  |    11 |   143 |     3   (0)| 00:00:01 |
-----------------------------------------------------------------------------
SQL> SET AUTOTRACE OFF
```

- *집합 연산자 사용 시 주의 사항*

```
SQL> SELECT deptno, empno, ename, sal
       FROM emp
       WHERE deptno IN (10,30)
       ORDER BY ename
       UNION
       SELECT deptno, empno, ename, sal
       FROM emp
       WHERE deptno IN (20,30)
       ORDER BY ename ;
ERROR at line 5:
ORA-00933: SQL command not properly ended
```

```
SQL> SELECT deptno, empno, ename, sal
       FROM emp
       WHERE deptno IN (10,30)
       UNION
       SELECT deptno, empno, ename, sal
       FROM emp
       WHERE deptno IN (20,30)
       ORDER BY ename ;
    DEPTNO      EMPNO ENAME             SAL
---------- ---------- ---------- ----------
        20       7876 ADAMS            1100
        30       7499 ALLEN            1600
        30       7698 BLAKE            2850
        10       7782 CLARK            2450
        20       7902 FORD             3000
        30       7900 JAMES             950
        20       7566 JONES            2975
        10       7839 KING             5000
        30       7654 MARTIN           1250
        10       7934 MILLER           1300
        20       7788 SCOTT            3000
        20       7369 SMITH             800
        30       7844 TURNER           1500
        30       7521 WARD             1250
14 rows selected.
SQL> SELECT deptno, SUM(sal)
       FROM emp
       GROUP BY deptno
       UNION ALL
       SELECT job, SUM(sal)
       FROM emp
       GROUP BY job ;
ERROR at line 1:
ORA-01790: expression must have same datatype as corresponding expression
```

```
SQL> SELECT deptno, NULL AS JOB, SUM(sal)
       FROM emp
       GROUP BY deptno
       UNION ALL
       SELECT NULL, job, SUM(sal)
       FROM emp
       GROUP BY job
       ORDER BY deptno ;
    DEPTNO JOB         SUM(SAL)
---------- --------- ----------
        10 (NULL)          8750
        20 (NULL)         10875
        30 (NULL)          9400
(NULL)     MANAGER         8275
(NULL)     PRESIDENT       5000
(NULL)     SALESMAN        5600
(NULL)     ANALYST         6000
(NULL)     CLERK           4150
```

Quiz

1. EMP 테이블에서 deptno, job 으로 그룹화 된 급여의 합계와 deptno, mgr 로 그룹화 된 급여의 합계를 다음과 같이 표시 하시오.

```
    DEPTNO JOB              MGR   SUM(SAL)
---------- --------- ---------- ----------
        10 CLERK     (NULL)           1300
        10 MANAGER   (NULL)           2450
        10 PRESIDENT (NULL)           5000
        10 (NULL)          7782       1300
        10 (NULL)          7839       2450
        10 (NULL)    (NULL)           5000
        20 ANALYST   (NULL)           6000
        20 CLERK     (NULL)           1900
        20 MANAGER   (NULL)           2975
        20 (NULL)          7566       6000
        20 (NULL)          7788       1100
        20 (NULL)          7839       2975
        20 (NULL)          7902        800
        30 CLERK     (NULL)            950
        30 MANAGER   (NULL)           2850
        30 SALESMAN  (NULL)           5600
        30 (NULL)          7698       6550
        30 (NULL)          7839       2850
18 rows selected.
```

9. 데이터 조작

• *INSERT 명령문 사용*

```
SQL> DROP TABLE copy_emp PURGE ;
ERROR at line 1:
ORA-00942: table or view does not exist
SQL> CREATE TABLE copy_emp
        AS
        SELECT * FROM emp WHERE 1 = 0 ;
Table created.
SQL> SELECT * FROM copy_emp ;
no rows selected
SQL> INSERT INTO copy_emp ( empno, ename, job, mgr, hiredate, sal, comm, deptno )
        VALUES ( 7369, 'SMITH', 'CLERK', 7902, TO_DATE('80/12/17','RR/MM/DD') , 800, NULL, 20 ) ;
1 row created.
SQL> INSERT INTO copy_emp ( empno, ename, hiredate, deptno )
        VALUES ( 7499, 'ALLEN', SYSDATE, 30 ) ;
1 row created.
SQL> INSERT INTO copy_emp
       SELECT * FROM emp
       WHERE deptno = 10 ;
3 rows created.
SQL> SELECT * FROM copy_emp ;
     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800 (NULL)             20
      7499 ALLEN      (NULL)    (NULL)     14-MAY-13 (NULL)     (NULL)             30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450 (NULL)             10
      7839 KING       PRESIDENT (NULL)     17-NOV-81       5000 (NULL)             10
      7934 MILLER     CLERK           7782 23-JAN-82       1300 (NULL)             10
SQL> COMMIT ;
Commit complete.
```

• *UPDATE 명령문 사용*

```
SQL> UPDATE copy_emp
        SET sal = 5000
        WHERE empno = 7369 ;
1 row updated.
SQL> UPDATE copy_emp
        SET hiredate = SYSDATE , comm = NULL
        WHERE empno = 7369 ;
1 row updated.
SQL> UPDATE copy_emp
        SET deptno = 50 ;
5 rows updated.
```

```
SQL> SELECT * FROM copy_emp ;
     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 14-MAY-13       5000 (NULL)             50
      7499 ALLEN      (NULL)    (NULL)     14-MAY-13 (NULL)     (NULL)             50
      7782 CLARK      MANAGER         7839 09-JUN-81       2450 (NULL)             50
      7839 KING       PRESIDENT (NULL)     17-NOV-81       5000 (NULL)             50
      7934 MILLER     CLERK           7782 23-JAN-82       1300 (NULL)             50
SQL> ROLLBACK ;
Rollback complete.
SQL> SELECT * FROM copy_emp ;
     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800 (NULL)             20
      7499 ALLEN      (NULL)    (NULL)     14-MAY-13 (NULL)     (NULL)             30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450 (NULL)             10
      7839 KING       PRESIDENT (NULL)     17-NOV-81       5000 (NULL)             10
      7934 MILLER     CLERK           7782 23-JAN-82       1300 (NULL)             10

SQL> UPDATE copy_emp
       SET hiredate = ( SELECT hiredate FROM emp
                        WHERE empno = 7499 )
       WHERE empno = 7499 ;
1 row updated.
SQL> UPDATE copy_emp
       SET (job, mgr, sal, comm) = ( SELECT job, mgr, sal, comm
                                     FROM emp
                                     WHERE empno = 7499 )
       WHERE empno = 7499 ;
1 row updated.
SQL> SELECT * FROM copy_emp
       WHERE empno = 7499 ;
EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
SQL> COMMIT ;
Commit complete.
```

• *DELETE 명령문 사용*

```
SQL> DELETE copy_emp
       WHERE empno = 7369 ;
1 row deleted.
SQL> SELECT * FROM copy_emp ;
EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450 (NULL)             10
      7839 KING       PRESIDENT (NULL)     17-NOV-81       5000 (NULL)             10
      7934 MILLER     CLERK           7782 23-JAN-82       1300 (NULL)             10
```

```
SQL> DELETE copy_emp ;
4 rows deleted.
SQL> SELECT * FROM copy_emp ;
no rows selected
SQL> ROLLBACK ;
Rollback complete.
SQL> DELETE copy_emp
       WHERE deptno = ( SELECT deptno FROM emp
                        WHERE empno = 7839 ) ;
3 rows deleted.
SQL> SELECT * FROM copy_emp ;
EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800 (NULL)             20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
SQL> ROLLBACK ;
Rollback complete.
```

- *TRUNCATE 명령문 사용*

```
SQL> DELETE copy_emp ;
5 rows deleted.
SQL> SELECT * FROM copy_emp ;
no rows selected
SQL> ROLLBACK ;
Rollback complete.
SQL> SELECT * FROM copy_emp ;
EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80       7000 (NULL)             20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450 (NULL)             10
      7839 KING       PRESIDENT (NULL)     17-NOV-81       5000 (NULL)             10
      7934 MILLER     CLERK           7782 23-JAN-82       1300 (NULL)             10
SQL> SELECT * FROM salgrade ;
GRADE      LOSAL      HISAL
---------- ---------- ----------
         1        700       1200
         2       1201       1400
         3       1401       2000
         4       2001       3000
         5       3001       9999

SQL> TRUNCATE TABLE salgrade ;
Table truncated.
SQL> SELECT * FROM salgrade ;
no rows selected
SQL> ROLLBACK;
Rollback complete.
SQL> SELECT * FROM salgrade ;
no rows selected
```

- *SAVEPOINT 사용*

```
SQL> UPDATE copy_emp
       SET sal = 6000
       WHERE empno = 7369 ;
1 row updated.
SQL> SAVEPOINT update_done ;
Savepoint created.
SQL> DELETE copy_emp
       WHERE empno = 7499 ;
1 row deleted.
SQL> ROLLBACK TO update_done ;
Rollback complete.
SQL> SELECT * from copy_emp ;
EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        6000 (NULL)            20
      7499 ALLEN      SALESMAN        7698 20-FEB-81        1600        300        30
      7782 CLARK      MANAGER         7839 09-JUN-81        2450 (NULL)            10
      7839 KING       PRESIDENT (NULL)      17-NOV-81        5000 (NULL)            10
      7934 MILLER     CLERK           7782 23-JAN-82        1300 (NULL)            10
SQL> ROLLBACK ;
Rollback complete.
SQL> SELECT * from copy_emp ;
EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80         800 (NULL)            20
      7499 ALLEN      SALESMAN        7698 20-FEB-81        1600        300        30
      7782 CLARK      MANAGER         7839 09-JUN-81        2450 (NULL)            10
      7839 KING       PRESIDENT (NULL)      17-NOV-81        5000 (NULL)            10
      7934 MILLER     CLERK           7782 23-JAN-82        1300 (NULL)            10
```

- *트랜잭션 관리의 주의 사항 확인*

```
SQL> UPDATE copy_emp
       SET sal = 7000
       WHERE empno = 7369 ;
1 row updated.
SQL> ALTER TABLE copy_emp
       MODIFY (sal    NUMBER(8,2)) ;
Table altered.
SQL> SELECT * FROM copy_emp ;
EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        7000 (NULL)            20
      7499 ALLEN      SALESMAN        7698 20-FEB-81        1600        300        30
      7782 CLARK      MANAGER         7839 09-JUN-81        2450 (NULL)            10
      7839 KING       PRESIDENT (NULL)      17-NOV-81        5000 (NULL)            10
      7934 MILLER     CLERK           7782 23-JAN-82        1300 (NULL)            10
SQL> ROLLBACK ;
Rollback complete.
```

```
SQL> SELECT * FROM copy_emp ;
EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80       7000 (NULL)             20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600         300        30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450 (NULL)             10
      7839 KING       PRESIDENT (NULL)     17-NOV-81       5000 (NULL)             10
      7934 MILLER     CLERK           7782 23-JAN-82       1300 (NULL)             10
```

- *읽기 일관성 확인*

※ 두개의 터미널을 실행하여 순서대로 명령문을 실행

Terminal 1	**Terminal 2**
$ sqlplus ora1/oracle SQL> SELECT empno, ename, sal FROM copy_emp WHERE deptno = 10 ; EMPNO ENAME SAL ---------- ---------- ---------- 7782 CLARK 2450 7839 KING 5000 7934 MILLER 1300	$ sqlplus ora1/oracle SQL> SELECT empno, ename, sal FROM copy_emp WHERE deptno = 10 ; EMPNO ENAME SAL ---------- ---------- ---------- 7782 CLARK 2450 7839 KING 5000 7934 MILLER 1300

SQL> UPDATE copy_emp
SET sal = 9999
WHERE empno = 7839 ;
1 row updated.	

SQL> SELECT empno, ename, sal FROM copy_emp WHERE deptno = 10 ; EMPNO ENAME SAL ---------- ---------- ---------- 7782 CLARK 2450 7839 KING 9999 7934 MILLER 1300	SQL> SELECT empno, ename, sal FROM copy_emp WHERE deptno = 10 ; EMPNO ENAME SAL ---------- ---------- ---------- 7782 CLARK 2450 7839 KING 5000 7934 MILLER 1300

	SQL> UPDATE copy_emp
	SET sal = 3000
	WHERE empno = 7934 ;
	1 row updated.

SQL> SELECT empno, ename, sal FROM copy_emp WHERE deptno = 10 ; EMPNO ENAME SAL ---------- ---------- ---------- 7782 CLARK 2450 7839 KING 9999 7934 MILLER 1300	SQL> SELECT empno, ename, sal FROM copy_emp WHERE deptno = 10 ; EMPNO ENAME SAL ---------- ---------- ---------- 7782 CLARK 2450 7839 KING 5000 7934 MILLER 3000

	SQL> DROP TABLE copy_emp ; ERROR at line 1: ORA-00054: resource busy and acquire with NOWAIT specified or timeout expired SQL> UPDATE copy_emp SET sal = 8000 WHERE empno = 7839 ; *※ Waiting*
SQL> ROLLBACK ; Rollback complete.	1 row updated. SQL> ROLLBACK ; Rollback complete.

※ 실습 종료 후 터미널은 모두 종료

- *Flashback Query, Flashback Version Query 사용*

```
SQL> SELECT TO_CHAR(SYSDATE,'YYYY/MM/DD HH24:MI:SS') FROM dual ;
TO_CHAR(SYSDATE,'YY
-------------------
2013/05/14 16:15:58
SQL> UPDATE emp
       SET sal = sal * 1.2
       WHERE empno = 7788 ;
1 row updated.
SQL> COMMIT ;
Commit complete.
SQL> SELECT empno, sal
       FROM emp
       WHERE empno = 7788 ;
     EMPNO        SAL
---------- ----------
      7788       3600
SQL> SELECT empno, sal
       FROM emp AS OF TIMESTAMP TO_DATE('2013/05/14 16:15:58','YYYY/MM/DD HH24:MI:SS')
       WHERE empno = 7788 ;
     EMPNO        SAL
---------- ----------
      7788       3000
SQL> SELECT empno, sal
       FROM emp AS OF TIMESTAMP ( SYSDATE - INTERVAL '10' MINUTE )
       WHERE empno = 7788 ;
     EMPNO        SAL
---------- ----------
      7788       3000
```

```
SQL> UPDATE emp
        SET sal = sal * 1.2
        WHERE empno = 7788 ;
1 row updated.
SQL> COMMIT ;
Commit complete.
SQL> SELECT empno, sal
        FROM emp
        WHERE empno = 7788 ;
EMPNO             SAL
---------- ----------
      7788       4320
SQL> SELECT empno, sal, versions_starttime, versions_endtime, versions_startscn, versions_endscn
        FROM emp VERSIONS BETWEEN TIMESTAMP MINVALUE AND MAXVALUE
        WHERE empno = 7788 ;
EMPNO             SAL VERSIONS_STARTTIME        VERSIONS_ENDTIME          VERSIONS_STARTSCN VERSIONS_ENDSCN
---------- ---------- ------------------------- ------------------------- ----------------- ---------------
      7788       4320 14-MAY-13 04.18.13 PM     (NULL)                              1294309 (NULL)
      7788       3600 14-MAY-13 04.16.18 PM     14-MAY-13 04.18.13 PM               1294258         1294309
      7788       3000 (NULL)                    14-MAY-13 04.16.18 PM     (NULL)                    1294258

SQL> UPDATE emp
        SET sal = ( SELECT sal
                    FROM emp AS OF SCN 1294257
                    WHERE empno = 7788 )
        WHERE empno = 7788 ;
1 row updated.

SQL> SELECT empno, sal
        FROM emp
        WHERE empno = 7788 ;
EMPNO             SAL
---------- ----------
      7788       3000
SQL> COMMIT ;
Commit complete.
```

10. DDL 문을 사용하여 테이블 생성 및 관리

- *문자 타입 확인*

```
SQL> CREATE TABLE t1
       ( c1      CHAR(5),
         c2      VARCHAR2(5),
         c3      LONG,
         c4      CLOB ) ;
Table created.
SQL> DESC t1
Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 C1                                                 CHAR(5)
 C2                                                 VARCHAR2(5)
 C3                                                 LONG
 C4                                                 CLOB

SQL> INSERT INTO t1 (c1, c2,c3,c4)
       VALUES ('ABC','ABC', 'ABC', 'ABC') ;
1 row created.
SQL> SELECT c1, c2, c3, c4 FROM t1 ;
C1    C2    C3         C4
----- ----- ---------- ----------
ABC   ABC   ABC        ABC

SQL> SELECT LENGTH(c1), LENGTH(c2), DUMP(c1), DUMP(c2) FROM t1 ;
LENGTH(C1) LENGTH(C2) DUMP(C1)                       DUMP(C2)
---------- ---------- ------------------------------ ------------------------------
         5          3 Typ=96 Len=5: 65,66,67,32,32   Typ=1 Len=3: 65,66,67

SQL> SELECT c1, c2
       FROM t1
       WHERE c2 = 'ABC' ; ;
C1    C2
----- -----
ABC   ABC
SQL> SELECT c1, c2
       FROM t1
       WHERE c1 = 'ABC' ;
C1    C2
----- -----
ABC   ABC
SQL> SELECT c1, c2
       FROM t1
       WHERE c1 = 'ABC   ' ;
C1    C2
----- -----
ABC   ABC
```

```
SQL> SELECT c1, c2
       FROM t1
       WHERE c1 = 'ABC        ' ;
C1    C2
----- -----
ABC   ABC
```

```
SQL> SELECT c1, c2
       FROM t1
       WHERE c1 = c2 ;
no rows selected
SQL> SELECT c1, c2
       FROM t1
       WHERE TRIM(c1) = c2 ;
```

• LONG Type & LOB Type 확인

```
SQL> ALTER TABLE t1
       ADD (c5   LONG) ;
ERROR at line 2:
ORA-01754: a table may contain only one column of type LONG
```

```
SQL> SELECT table_name, column_name, segment_name
       FROM user_lobs ;
TABLE_NAME      COLUMN_NAME     SEGMENT_NAME
--------------- --------------- ------------------------------
T1              C4              SYS_LOB0000079784C00004$$
```

```
SQL> ALTER TABLE t1
       ADD (c5   CLOB) ;
Table altered.
SQL> SELECT table_name, column_name, segment_name
       FROM user_lobs ;
TABLE_NAME      COLUMN_NAME     SEGMENT_NAME
--------------- --------------- ------------------------------
T1              C4              SYS_LOB0000079784C00004$$
T1              C5              SYS_LOB0000079784C00005$$
```

```
SQL> SELECT object_name, object_type
       FROM user_objects
       WHERE object_name LIKE 'SYS%' ;
OBJECT_NAME                    OBJECT_TYPE
------------------------------ -------------------
SYS_LOB0000079784C00005$$      LOB
SYS_LOB0000079784C00004$$      LOB
```

- *숫자 타입 확인*

```
SQL> CREATE TABLE t2
       ( c1       NUMBER,
         c2       NUMBER(4,2),
         c3       NUMBER(4,0),
         c4       NUMBER(2,4),
         c5       NUMBER(3,-1) ) ;
Table created.
SQL> INSERT INTO t2
       VALUES (999999.999999 , 99.99, 9999, 0.0099, 9990) ;
1 row created.
SQL> INSERT INTO t2 (c2)
       VALUES (999.99) ;
ERROR at line 2:
ORA-01438: value larger than specified precision allowed for this column
SQL> INSERT INTO t2 (c3)
       VALUES (99999) ;
ERROR at line 2:
ORA-01438: value larger than specified precision allowed for this column
SQL> INSERT INTO t2 (c4)
       VALUES (0.0999) ;
ERROR at line 2:
ORA-01438: value larger than specified precision allowed for this column
SQL> INSERT INTO t2 (c5)
       VALUES (99990) ;
ERROR at line 2:
ORA-01438: value larger than specified precision allowed for this column
```

- *날짜 타입 확인*

```
SQL> CREATE TABLE t3
        ( c1      DATE,
          c2      TIMESTAMP,
          c3      TIMESTAMP WITH TIME ZONE,
          c4      TIMESTAMP WITH LOCAL TIME ZONE,
          c5      INTERVAL YEAR TO MONTH,
          c6      INTERVAL DAY TO SECOND ) ;
Table created.
SQL> SELECT dbtimezone, sessiontimezone FROM dual ;
DBTIME SESSIONTIMEZONE
------ ---------------------------------------------------------------------------
+00:00 +09:00

SQL> ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MON-RR HH24:MI:SS' ;
Session altered.
```

```
SQL> SELECT SYSDATE, SYSTIMESTAMP FROM dual ;
SYSDATE            SYSTIMESTAMP
------------------ ---------------------------------------------------------------------------
09-MAY-13 00:00:55 09-MAY-13 12.00.55.813603 AM +09:00
SQL> SELECT CURRENT_DATE, CURRENT_TIMESTAMP FROM dual ;
CURRENT_DATE       CURRENT_TIMESTAMP
------------------ ---------------------------------------------------------------------------
09-MAY-13 00:01:25 09-MAY-13 12.01.25.000000 AM +09:00


SQL> ALTER SESSION SET TIME_ZONE = '-5:00' ;
Session altered.
SQL> SELECT dbtimezone, sessiontimezone FROM dual ;
DBTIME SESSIONTIMEZONE
------ ---------------------------------------------------------------------------
+00:00 -05:00
SQL> SELECT SYSDATE, SYSTIMESTAMP FROM dual ;
SYSDATE            SYSTIMESTAMP
------------------ ---------------------------------------------------------------------------
09-MAY-13 00:03:26 09-MAY-13 12.03.26.058698 AM +09:00
SQL> SELECT CURRENT_DATE, CURRENT_TIMESTAMP FROM dual ;
CURRENT_DATE       CURRENT_TIMESTAMP
------------------ ---------------------------------------------------------------------------
08-MAY-13 10:03:36 08-MAY-13 10.03.36.000000 AM -05:00
SQL> INSERT INTO t3 (c1,c2,c3,c4)
       VALUES (SYSTIMESTAMP, SYSTIMESTAMP, SYSTIMESTAMP, SYSTIMESTAMP) ;
1 row created.
SQL> SELECT c1,c2,c3,c4 FROM t3 ;
C1                 C2                             C3                                  C4
------------------ ------------------------------ ----------------------------------- ------------------------------
09-MAY-13 00:04:42 09-MAY-13 12.04.42.787924 AM   09-MAY-13 12.04.42.787924 AM +09:00 08-MAY-13 10.04.42.787924 AM
SQL> ALTER SESSION SET TIME_ZONE = '-10:00' ;
Session altered.
SQL> SELECT c1,c2,c3,c4 FROM t3 ;
C1                 C2                             C3                                  C4
------------------ ------------------------------ ----------------------------------- ------------------------------
09-MAY-13 00:04:42 09-MAY-13 12.04.42.787924 AM   09-MAY-13 12.04.42.787924 AM +09:00 08-MAY-13 05.04.42.787924 AM


SQL> UPDATE t3
       SET c5 = '1-5', c6 = '5 15:11:10' ;
1 row updated.
SQL> SELECT c1, c5, c6, c1 + c5, c1 + c6 FROM t3 ;
C1                 C5         C6                   C1+C5              C1+C6
------------------ ---------- -------------------- ------------------ ------------------
09-MAY-13 00:04:42 +01-05     +05 15:11:10.000000  09-OCT-14 00:04:42 14-MAY-13 15:15:52


SQL> SELECT SYSDATE, SYSDATE + TO_YMINTERVAL ('1-2')
       FROM dual ;
SYSDATE            SYSDATE+TO_YMINTER
------------------ ------------------
09-MAY-13 00:09:57 09-JUL-14 00:09:57
```

```
SQL> SELECT SYSDATE, SYSDATE + TO_DSINTERVAL('5 10:10:15')
       FROM dual ;
SYSDATE            SYSDATE+TO_DSINTER
------------------ ------------------
09-MAY-13 00:10:32 14-MAY-13 10:20:47


SQL> CREATE TABLE t4
        ( c1      RAW(2000),
          c2      LONG RAW,
          c3      BLOB,
          c4      BFILE ) ;
Table created.
SQL> DROP TABLE t1 PURGE ;
Table dropped.
SQL> DROP TABLE t2 PURGE ;
Table dropped.
SQL> DROP TABLE t3 PURGE ;
Table dropped.
SQL> DROP TABLE t4 PURGE ;
Table dropped.
SQL> ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MON-RR' ;
Session altered.
```

- *DEFAULT 옵션 확인*

```
SQL> SELECT table_name, column_name, data_type,   data_default
       FROM user_tab_columns
       WHERE table_name = 'EMP' ;
TABLE_NAME      COLUMN_NAME     DATA_TYPE  DATA_DEFAULT
--------------- --------------- ---------- --------------------------
EMP             EMPNO           NUMBER     (NULL)
EMP             ENAME           VARCHAR2   (NULL)
EMP             JOB             VARCHAR2   (NULL)
EMP             MGR             NUMBER     (NULL)
EMP             HIREDATE        DATE       (NULL)
EMP             SAL             NUMBER     (NULL)
EMP             COMM            NUMBER     (NULL)
EMP             DEPTNO          NUMBER     (NULL)


SQL> INSERT INTO emp (empno, ename, deptno)
       VALUES (1234, 'RYU',30) ;
1 row created.
SQL> SELECT empno, ename, hiredate, deptno
       FROM emp
       WHERE empno = 1234 ;
     EMPNO ENAME      HIREDATE               DEPTNO
---------- ---------- ------------------ ----------
      1234 RYU        (NULL)                     30
SQL> ROLLBACK ;
Rollback complete.
```

```
SQL> ALTER TABLE emp
        MODIFY (hiredate   DATE DEFAULT SYSDATE) ;
Table altered.

SQL> SELECT table_name, column_name, data_type,   data_default
        FROM user_tab_columns
        WHERE table_name = 'EMP' ;
TABLE_NAME      COLUMN_NAME     DATA_TYPE  DATA_DEFAULT
--------------- --------------- ---------- ----------------------
EMP             EMPNO           NUMBER     (NULL)
EMP             ENAME           VARCHAR2   (NULL)
EMP             JOB             VARCHAR2   (NULL)
EMP             MGR             NUMBER     (NULL)
EMP             HIREDATE        DATE       SYSDATE
EMP             SAL             NUMBER     (NULL)
EMP             COMM            NUMBER     (NULL)
EMP             DEPTNO          NUMBER     (NULL)

SQL> INSERT INTO emp (empno, ename, deptno)
        VALUES (1234, 'RYU',30) ;
1 row created.
SQL> SELECT empno, ename, hiredate, deptno
        FROM emp
        WHERE empno = 1234 ;
     EMPNO ENAME      HIREDATE      DEPTNO
---------- ---------- --------- ----------
      1234 RYU        09-MAY-13         30
SQL> UPDATE emp
        SET hiredate = DEFAULT
        WHERE empno = 1234 ;
1 row updated.
SQL> ROLLBACK ;
Rollback complete.
```

- *제약 조건 확인*

```
SQL> SELECT constraint_name, table_name, constraint_type, search_condition
        FROM user_constraints
        WHERE table_name IN ('EMP','DEPT') ;
no rows selected
SQL> UPDATE emp
        SET empno = 1234,
            ename = 'RYU',
            sal     = -1000,
            deptno = 50 ;
14 rows updated.
```

```
SQL> SELECT empno, ename, sal, deptno FROM emp ;
EMPNO ENAME             SAL      DEPTNO
---------- ---------- ---------- ----------
      1234 RYU             -1000         50
      1234 RYU             -1000         50
      1234 RYU             -1000         50
      1234 RYU             -1000         50
...
SQL> ROLLBACK ;
Rollback complete.


SQL> ALTER TABLE dept
       ADD CONSTRAINT dept_pk PRIMARY KEY (deptno) ;
Table altered.
SQL> ALTER TABLE emp
       ADD CONSTRAINT emp_pk PRIMARY KEY(empno) ;
Table altered.
SQL> ALTER TABLE emp
       MODIFY ename NOT NULL ;
Table altered.
SQL> ALTER TABLE emp
       ADD UNIQUE (ename) ;
Table altered.
SQL> ALTER TABLE emp
       ADD CONSTRAINT emp_ck CHECK (SAL > 0) ;
Table altered.
SQL> ALTER TABLE emp
       ADD CONSTRAINT emp_fk FOREIGN KEY (deptno) REFERENCES dept(deptno) ;
Table altered.
SQL> SELECT constraint_name, table_name, constraint_type, search_condition
       FROM user_constraints
       WHERE table_name IN ('EMP','DEPT') ;
CONSTRAINT_NAME                TABLE_NAME      C SEARCH_CONDITION
------------------------------ --------------- - ------------------------------
DEPT_PK                        DEPT            P (NULL)
EMP_PK                         EMP             P (NULL)
SYS_C0015569                   EMP             C "ENAME" IS NOT NULL
EMP_CK                         EMP             C SAL > 0
EMP_FK                         EMP             R (NULL)
SYS_C0015572                   EMP             U (NULL)


SQL> UPDATE emp
       SET empno = 7788 ;
ERROR at line 1:
ORA-00001: unique constraint (ORA1.EMP_PK) violated
SQL> UPDATE emp
       SET ename = NULL ;
ERROR at line 2:
ORA-01407: cannot update ("ORA1"."EMP"."ENAME") to NULL
```

```
SQL> UPDATE emp
       SET ename = 'RYU' ;
ERROR at line 1:
ORA-00001: unique constraint (ORA1.SYS_C0015572) violated
SQL> UPDATE emp
       SET sal = -1000 ;
ERROR at line 1:
ORA-02290: check constraint (ORA1.EMP_CK) violated
SQL> UPDATE emp
       SET deptno = 50 ;
ERROR at line 1:
ORA-02291: integrity constraint (ORA1.EMP_FK) violated - parent key not found
SQL> DELETE dept
       WHERE deptno = 40 ;
1 row deleted.
SQL> DELETE dept
       WHERE deptno = 10 ;
ERROR at line 1:
ORA-02292: integrity constraint (ORA1.EMP_FK) violated - child record found
SQL> ROLLBACK ;
Rollback complete.

SQL> ALTER TABLE emp
       DISABLE CONSTRAINT emp_fk ;
Table altered.
SQL> UPDATE emp
       SET deptno = 50 ;
14 rows updated.
SQL> ROLLBACK ;
Rollback complete.
SQL> ALTER TABLE emp
       ENABLE CONSTRAINT emp_fk ;
Table altered.

SQL> ALTER TABLE emp
       DROP CONSTRAINT emp_fk ;
Table altered.
SQL> ALTER TABLE emp
       DROP CONSTRAINT SYS_C0015572 ;
Table altered.
SQL> ALTER TABLE emp
       DROP CONSTRAINT emp_ck ;
Table altered.
SQL> ALTER TABLE emp
       DROP CONSTRAINT SYS_C0015569 ;
Table altered.
```

```
SQL> ALTER TABLE emp
        DROP PRIMARY KEY ;
Table altered.
SQL> ALTER TABLE dept
        DROP PRIMARY KEY ;
Table altered.
```

- *테이블 수정*

```
SQL> DROP TABLE copy_emp PURGE ;
Table dropped.
SQL> CREATE TABLE copy_emp
        AS SELECT * FROM emp ;
Table created.
SQL> ALTER TABLE copy_emp
        ADD ( dname       VARCHAR2(10) ) ;
Table altered.
SQL> ALTER TABLE copy_emp
        MODIFY ( dname    VARCHAR2(20) ) ;
Table altered.
SQL> ALTER TABLE copy_emp
        DROP COLUMN dname ;
Table altered.
SQL> ALTER TABLE copy_emp
        SET UNUSED COLUMN hiredate ;
Table altered.
SQL> SELECT *
        FROM user_unused_col_tabs ;
TABLE_NAME           COUNT
--------------- ----------
COPY_EMP                 1
SQL> ALTER TABLE copy_emp
        DROP UNUSED COLUMNS ;
Table altered.
SQL> ALTER TABLE copy_emp
        RENAME COLUMN ename TO name ;
Table altered.
SQL> ALTER TABLE copy_emp READ ONLY ;
Table altered.
SQL> ALTER TABLE copy_emp READ WRITE ;
Table altered.
```

- *테이블 삭제*

```
SQL> DROP TABLE copy_emp ;
Table dropped.
SQL> SELECT object_name, original_name, type, droptime
       FROM user_recyclebin ;
OBJECT_NAME                    ORIGINAL_NAME                    TYPE                      DROPTIME
------------------------------ -------------------------------- ------------------------- -------------------
BIN$3DgcCzJL3vTgQAB/AQAUQw==$0 COPY_EMP                         TABLE                     2013-05-09:01:05:58

SQL> SELECT empno, name, sal, deptno
       FROM "BIN$3DgcCzJL3vTgQAB/AQAUQw==$0" ;
     EMPNO NAME              SAL     DEPTNO
---------- ---------- ---------- ----------
      7369 SMITH             800         20
      7499 ALLEN            1600         30
      7521 WARD             1250         30
      7566 JONES            2975         20
      7654 MARTIN           1250         30
      7698 BLAKE            2850         30
...
SQL> FLASHBACK TABLE copy_emp TO BEFORE DROP RENAME TO cp_emp ;
Flashback complete.
SQL> SELECT empno, name, sal, deptno
       FROM cp_emp ;
     EMPNO NAME              SAL     DEPTNO
---------- ---------- ---------- ----------
      7369 SMITH             800         20
      7499 ALLEN            1600         30
      7521 WARD             1250         30
      7566 JONES            2975         20
      7654 MARTIN           1250         30
      7698 BLAKE            2850         30
...
SQL> DROP TABLE cp_emp ;
Table dropped.
SQL> PURGE RECYCLEBIN ;
Recyclebin purged.
```

- *임시 테이블 생성*

```
SQL> CREATE GLOBAL TEMPORARY TABLE temp1
       ( id         NUMBER,
         name       VARCHAR2(10))
       ON COMMIT DELETE ROWS ;
Table created.
SQL> CREATE GLOBAL TEMPORARY TABLE temp2
       ( id         NUMBER,
         name       VARCHAR2(10))
       ON COMMIT PRESERVE ROWS ;
Table created.
```

```
SQL> INSERT INTO temp1
       VALUES (1111, 'AAA') ;
1 row created.
SQL> INSERT INTO temp2
       VALUES (2222,'BBB') ;
1 row created.
SQL> SELECT * FROM temp1 ;
        ID NAME
---------- ----------
      1111 AAA
SQL> SELECT * FROM temp2 ;
        ID NAME
---------- ----------
      2222 BBB
SQL> COMMIT ;
Commit complete.
SQL> SELECT * FROM temp1 ;
no rows selected
SQL> SELECT * FROM temp2 ;
        ID NAME
---------- ----------
      2222 BBB
SQL> connect ora1/oracle
Connected.
SQL> SELECT * FROM temp2 ;
no rows selected
SQL> DROP TABLE temp1 PURGE ;
Table dropped.
SQL> DROP TABLE temp2 PURGE ;
Table dropped.
```

- *External Table 확인*

```
SQL> conn system/oracle
Connected.
SQL> CREATE OR REPLACE DIRECTORY data_dir AS '/home/oracle' ;
Directory created.
SQL> GRANT read, write ON DIRECTORY data_dir TO ora1 ;
Grant succeeded.
SQL> conn ora1/oracle
Connected.
SQL> host
$ cat >> /home/oracle/emp.dat << EOF
10,jones,11-DEC-1934
20,smith,12-JUN-1972
EOF
$ exit
```

```
SQL> CREATE TABLE oldemp
       ( deptno    NUMBER(2),
         ename      VARCHAR2(10),
         hiredate    DATE)
       ORGANIZATION EXTERNAL
       (TYPE ORACLE_LOADER
         DEFAULT DIRECTORY data_dir
         ACCESS PARAMETERS
         (RECORDS DELIMITED BY NEWLINE
           NOBADFILE
           NOLOGFILE
           FIELDS TERMINATED BY ','
           (deptno,ename,hiredate CHAR DATE_FORMAT DATE MASK "DD-MON-RR"))
         LOCATION ('emp.dat')) ;
Table created.
SQL> SELECT * FROM oldemp ;
DEPTNO ENAME      HIREDATE
---------- ---------- ---------
        10 jones      11-DEC-34
        20 smith      12-JUN-72
SQL> UPDATE oldemp
       SET deptno = 30
       WHERE ename = 'jones' ;
ERROR at line 1:
ORA-30657: operation not supported on external organized table
SQL> CREATE TABLE emp_ext
       (employee_id, first_name, last_name)
       ORGANIZATION EXTERNAL
       (TYPE ORACLE_DATAPUMP
         DEFAULT DIRECTORY data_dir
         LOCATION ('emp1.exp','emp2.exp'))
       PARALLEL
       AS SELECT employee_id, first_name, last_name
            FROM employees;
Table created.
SQL> SELECT * FROM emp_ext ;
EMPLOYEE_ID FIRST_NAME           LAST_NAME
----------- -------------------- -------------------------
        200 Jennifer             Whalen
        201 Michael              Hartstein
        202 Pat                  Fay
...
20 rows selected.
SQL> ! ls -al /home/oracle/emp*
-rw-r----- 1 oracle oinstall 12288 May 14 16:36 /home/oracle/emp1.exp
-rw-r----- 1 oracle oinstall 12288 May 14 16:36 /home/oracle/emp2.exp
```

11. 기타 스키마 객체 생성

- *View 생성*

```
SQL> SELECT empno, ename, sal, deptno
       FROM emp
       WHERE deptno = 10 ;
     EMPNO ENAME             SAL     DEPTNO
---------- ---------- ---------- ----------
      7782 CLARK            2450         10
      7839 KING             5000         10
      7934 MILLER           1300         10
SQL> CREATE VIEW empv10
       AS
       SELECT empno, ename, sal, deptno
       FROM emp
       WHERE deptno = 10 ;
View created.
SQL> DESC empv10
Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPNO                                              NUMBER(4)
 ENAME                                              VARCHAR2(10)
 SAL                                                NUMBER(7,2)
 DEPTNO                                             NUMBER(2)

SQL> SELECT * FROM empv10 ;
     EMPNO ENAME             SAL     DEPTNO
---------- ---------- ---------- ----------
      7782 CLARK            2450         10
      7839 KING             5000         10
      7934 MILLER           1300         10
SQL> UPDATE empv10
       SET sal = sal * 1.2
       WHERE empno = 7839 ;
1 row updated.
SQL> SELECT * FROM empv10 ;
     EMPNO ENAME             SAL     DEPTNO
---------- ---------- ---------- ----------
      7782 CLARK            2450         10
      7839 KING             6000         10
      7934 MILLER           1300         10
SQL> UPDATE empv10
       SET deptno = 20
       WHERE empno = 7934 ;
1 row updated.
SQL> SELECT * FROM empv10 ;
     EMPNO ENAME             SAL     DEPTNO
---------- ---------- ---------- ----------
      7782 CLARK            2450         10
      7839 KING             6000         10
```

```
SQL> SELECT empno, ename, sal, deptno
        FROM emp
        WHERE empno IN (7839, 7934) ;
     EMPNO ENAME             SAL     DEPTNO
---------- ---------- ---------- ----------
      7839 KING             6000         10
      7934 MILLER           1300         20
SQL> ROLLBACK ;
Rollback complete.

SQL> SELECT view_name, text
        FROM user_views
        WHERE view_name = 'EMPV10' ;
VIEW_NAME                      TEXT
------------------------------ --------------------------------------------------------------------------------
EMPV10                         SELECT empno, ename, sal, deptno
                               FROM emp
                               WHERE deptno = 10
SQL> SELECT *
        FROM (SELECT empno, ename, sal, deptno
                FROM emp
                WHERE deptno = 10) ;
     EMPNO ENAME             SAL     DEPTNO
---------- ---------- ---------- ----------
      7782 CLARK            2450         10
      7839 KING             5000         10
      7934 MILLER           1300         10

SQL> CREATE VIEW empv_sum
        AS
        SELECT deptno, SUM(sal)
        FROM emp
        GROUP BY deptno ;
ERROR at line 3:
ORA-00998: must name this expression with a column alias
SQL> CREATE VIEW empv_sum
        AS
        SELECT deptno, SUM(sal) AS SUM
        FROM emp
        GROUP BY deptno ;
View created.

SQL> SELECT * FROM empv_sum ;
    DEPTNO        SUM
---------- ----------
        30       9400
        20      10875
        10       8750
```

```
SQL> DELETE empv_sum
        WHERE deptno = 10 ;
ERROR at line 1:
ORA-01732: data manipulation operation not legal on this view
SQL> DROP VIEW empv_sum ;
View dropped.
SQL> CREATE VIEW empv10
        AS SELECT empno, ename, sal, comm, deptno
            FROM emp
            WHERE deptno = 10 ;
ERROR at line 1:
ORA-00955: name is already used by an existing object
SQL> CREATE OR REPLACE VIEW empv10
        AS SELECT empno, ename, sal, comm, deptno
            FROM emp
            WHERE deptno = 10 ;
View created.
SQL> CREATE OR REPLACE VIEW empv10
        AS SELECT empno, ename, sal, comm, deptno
            FROM emp
            WHERE deptno = 10 WITH CHECK OPTION ;
View created.
SQL> UPDATE empv10
        SET sal = 6000
        WHERE empno = 7839 ;
1 row updated.
SQL> UPDATE empv10
        SET deptno = 20
        WHERE empno = 7839 ;
ERROR at line 1:
ORA-01402: view WITH CHECK OPTION where-clause violation
SQL> ROLLBACK ;
Rollback complete.

SQL> CREATE OR REPLACE VIEW empv10
        AS SELECT empno, ename, sal, comm, deptno
            FROM emp
            WHERE deptno = 10 WITH READ ONLY ;
View created.
SQL> UPDATE empv10
        SET sal = 6000
        WHERE empno = 7839 ;
ERROR at line 2:
ORA-42399: cannot perform a DML operation on a read-only view
SQL> DROP VIEW empv10 ;
```

• Sequence 생성

```
SQL> CREATE SEQUENCE empno_seq
       START WITH 8000
       INCREMENT BY 1 ;
Sequence created.
SQL> SELECT sequence_name, increment_by, cache_size, last_number
       FROM user_sequences
       WHERE sequence_name = 'EMPNO_SEQ' ;
SEQUENCE_NAME                  INCREMENT_BY CACHE_SIZE LAST_NUMBER
------------------------------ ------------ ---------- -----------
EMPNO_SEQ                                 1         20        8000

SQL> INSERT INTO emp (empno, ename, deptno)
       VALUES (empno_seq.nextval, 'RYU',30) ;
1 row created.
SQL> INSERT INTO emp (empno, ename, deptno)
       VALUES (empno_seq.nextval, 'RYU',30) ;
1 row created.
SQL> SELECT empno, ename, deptno
       FROM emp
       WHERE ename = 'RYU' ;
     EMPNO ENAME          DEPTNO
---------- ---------- ----------
      8000 RYU                30
      8001 RYU                30

SQL> SELECT empno_seq.nextval FROM dual ;
   NEXTVAL
----------
      8002
SQL> SELECT empno_seq.nextval FROM dual ;
   NEXTVAL
----------
      8003
SQL> SELECT empno_seq.currval FROM dual ;
   CURRVAL
----------
      8003
SQL> SELECT empno_seq.currval FROM dual ;
   CURRVAL
----------
      8003

SQL> SELECT sequence_name, increment_by, cache_size, last_number
       FROM user_sequences
       WHERE sequence_name = 'EMPNO_SEQ' ;
SEQUENCE_NAME                  INCREMENT_BY CACHE_SIZE LAST_NUMBER
------------------------------ ------------ ---------- -----------
EMPNO_SEQ                                 1         20        8020
```

```
SQL> ROLLBACK ;
Rollback complete.
SQL> SELECT empno_seq.nextval FROM dual ;
   NEXTVAL
----------
      8004

SQL> CREATE SEQUENCE seq1
       START WITH 100
       INCREMENT BY 1
       MAXVALUE 103
       MINVALUE 50
       CYCLE
       NOCACHE ;
Sequence created.
SQL> SELECT seq1.nextval FROM dual ;
   NEXTVAL
----------
       100
SQL> SELECT seq1.nextval FROM dual ;
   NEXTVAL
----------
       101
SQL> SELECT seq1.nextval FROM dual ;
   NEXTVAL
----------
       102
SQL> SELECT seq1.nextval FROM dual ;
   NEXTVAL
----------
       103
SQL> SELECT seq1.nextval FROM dual ;
   NEXTVAL
----------
        50
SQL> SELECT seq1.nextval FROM dual ;
  NEXTVAL
----------
        51
SQL> ALTER SEQUENCE seq1
       MAXVALUE 200
       CACHE 20 ;
SQL> DROP SEQUENCE empno_seq ;
SQL> DROP SEQUENCE seq1 ;
```

- *인덱스 생성*

```
SQL> SELECT index_name, index_type, table_name, uniqueness
       FROM user_indexes
       WHERE table_name = 'EMP' ;
no rows selected
SQL> SET AUTOTRACE ON EXPLAIN
SQL> SELECT empno, ename, deptno
       FROM emp
       WHERE empno = 7788 ;
     EMPNO ENAME          DEPTNO
---------- ---------- ----------
      7788 SCOTT              20
Execution Plan
----------------------------------------------------------
Plan hash value: 3956160932
--------------------------------------------------------------------------
| Id  | Operation         | Name | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------
|   0 | SELECT STATEMENT  |      |     1 |    13 |     3   (0)| 00:00:01 |
|*  1 |  TABLE ACCESS FULL| EMP  |     1 |    13 |     3   (0)| 00:00:01 |
--------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------
   1 - filter("EMPNO"=7788)
SQL> SELECT empno, ename, deptno
       FROM emp
       WHERE ename = 'SCOTT' ;
     EMPNO ENAME          DEPTNO
---------- ---------- ----------
      7788 SCOTT              20
Execution Plan
----------------------------------------------------------
Plan hash value: 3956160932
--------------------------------------------------------------------------
| Id  | Operation         | Name | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------
|   0 | SELECT STATEMENT  |      |     1 |    13 |     3   (0)| 00:00:01 |
|*  1 |  TABLE ACCESS FULL| EMP  |     1 |    13 |     3   (0)| 00:00:01 |
--------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------
   1 - filter("ENAME"='SCOTT')
SQL> SET AUTOTRACE OFF
SQL> ALTER TABLE emp
       ADD PRIMARY KEY (empno) ;
Table altered.
SQL> SELECT index_name, index_type, table_name, uniqueness
       FROM user_indexes
       WHERE table_name = 'EMP' ;
INDEX_NAME                     INDEX_TYPE                  TABLE_NAME     UNIQUENES
------------------------------ --------------------------- -------------- ---------
SYS_C0015578                   NORMAL                      EMP            UNIQUE
```

```
SQL> CREATE INDEX emp_ename_ix ON emp(ename) ;
Index created.
SQL> SET AUTOTRACE ON EXPLAIN
SQL> SELECT empno, ename, deptno
       FROM emp
       WHERE empno = 7788 ;
     EMPNO ENAME          DEPTNO
---------- ---------- ----------
      7788 SCOTT              20
Execution Plan
----------------------------------------------------------
Plan hash value: 1843285278
--------------------------------------------------------------------------------------------
| Id  | Operation                   | Name         | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT            |              |     1 |    13 |     1   (0)| 00:00:01 |
|   1 |  TABLE ACCESS BY INDEX ROWID| EMP          |     1 |    13 |     1   (0)| 00:00:01 |
|*  2 |   INDEX UNIQUE SCAN         | SYS_C0015578 |     1 |       |     0   (0)| 00:00:01 |
--------------------------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------
   2 - access("EMPNO"=7788)
SQL> SELECT empno, ename, deptno
       FROM emp
       WHERE ename = 'SCOTT' ;
     EMPNO ENAME          DEPTNO
---------- ---------- ----------
      7788 SCOTT              20
Execution Plan
----------------------------------------------------------
Plan hash value: 80071485
--------------------------------------------------------------------------------------------
| Id  | Operation                   | Name         | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT            |              |     1 |    13 |     2   (0)| 00:00:01 |
|   1 |  TABLE ACCESS BY INDEX ROWID| EMP          |     1 |    13 |     2   (0)| 00:00:01 |
|*  2 |   INDEX RANGE SCAN          | EMP_ENAME_IX |     1 |       |     1   (0)| 00:00:01 |
--------------------------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------
   2 - access("ENAME"='SCOTT')

SQL> SET AUTOTRACE OFF
SQL> DROP INDEX emp_ename_ix ;
SQL> ALTER TABLE emp
       DROP PRIMARY KEY ;
```

- *Synonym 생성*

```
SQL> CONN system/oracle
SQL> SELECT empno, ename, deptno
       FROM emp ;
ERROR at line 2:
ORA-00942: table or view does not exist
```

```
SQL> SELECT empno, ename, deptno
       FROM ora1.emp ;
EMPNO ENAME          DEPTNO
---------- ---------- ----------
      7369 SMITH              20
      7499 ALLEN              30
      7521 WARD               30
      7566 JONES              20
...
SQL> CREATE SYNONYM emp FOR ora1.emp ;
SQL> SELECT empno, ename, deptno
       FROM emp ;
EMPNO ENAME          DEPTNO
---------- ---------- ----------
      7369 SMITH              20
      7499 ALLEN              30
      7521 WARD               30
      7566 JONES              20
...
SQL> SELECT synonym_name, table_owner, table_name
       FROM user_synonyms
       WHERE synonym_name = 'EMP' ;
SYNONYM_NAME                   TABLE_OWNER                    TABLE_NAME
------------------------------ ------------------------------ ---------------
EMP                            ORA1                           EMP
SQL> DROP SYNONYM emp ;
Synonym dropped.
SQL> conn ora1/oracle
Connected.
```

1. 유저 엑세스 제어

- *유저 생성*

```
$ sqlplus system/oracle
SQL> SELECT * FROM session_privs ;
PRIVILEGE
----------------------------------------
ALTER SYSTEM
AUDIT SYSTEM
CREATE SESSION
ALTER SESSION
RESTRICTED SESSION
CREATE TABLESPACE
ALTER TABLESPACE
MANAGE TABLESPACE
DROP TABLESPACE
UNLIMITED TABLESPACE
CREATE USER
...
SQL> CREATE USER user01
       IDENTIFIED BY oracle ;
User created.
SQL> CONN user01/oracle
ERROR:
ORA-01045: user USER01 lacks CREATE SESSION privilege; logon denied
Warning: You are no longer connected to ORACLE.
SQL> CONN system/oracle
Connected.
SQL> GRANT CREATE SESSION, CREATE TABLE, UNLIMITED TABLESPACE TO user01 ;
Grant succeeded.
SQL> CONN user01/oracle
Connected.
SQL> CREATE TABLE t1 ( id         number ) ;
Table created.
SQL> CREATE VIEW v1
       AS SELECT * FROM t1 ;
ERROR at line 1:
ORA-01031: insufficient privileges
```

- *시스템 권한 및 롤 관리*

```
SQL> CONN system/oracle
Connected.
SQL> CREATE ROLE manager ;
Role created.
SQL> GRANT CREATE SESSION, CREATE TABLE, CREATE VIEW, DBA TO manager ;
Grant succeeded.
SQL> REVOKE DBA FROM manager ;
Revoke succeeded.
SQL> GRANT manager TO user01 ;
Grant succeeded.
SQL> CONN user01/oracle
Connected.
```

```
SQL> SELECT * FROM session_roles ;
ROLE
------------------------------
MANAGER
SQL> SELECT * FROM session_privs ;
PRIVILEGE
----------------------------------------
CREATE SESSION
UNLIMITED TABLESPACE
CREATE TABLE
CREATE VIEW
```

- *객체 권한 관리*

```
SQL> SELECT empno, ename, deptno FROM ora1.emp ;
ERROR at line 1:
ORA-00942: table or view does not exist
SQL> CONN ora1/oracle
Connected.
SQL> GRANT select, update ON emp TO user01 ;
Grant succeeded.
SQL> GRANT insert ON emp TO manager ;
Grant succeeded.
SQL> GRANT delete ON emp TO user01 WITH GRANT OPTION ;
Grant succeeded.
SQL> GRANT select ON dept TO PUBLIC ;
Grant succeeded.

SQL> CONN user01/oracle
Connected.
SQL> SELECT empno, ename, deptno FROM ora1.emp ;
     EMPNO ENAME          DEPTNO
---------- ---------- ----------
      7369 SMITH              20
      7499 ALLEN              30
      7521 WARD               30
      7566 JONES              20
...
SQL> SELECT deptno, dname FROM ora1.dept ;
    DEPTNO DNAME
---------- --------------
        10 ACCOUNTING
        20 RESEARCH
        30 SALES
        40 OPERATIONS

SQL> GRANT select ON ora1.emp TO ora2 ;
ERROR at line 1:
ORA-01031: insufficient privileges
SQL> GRANT delete ON ora1.emp TO ora2 ;
Grant succeeded.
```

```
SQL> CONN ora2/oracle
Connected.
SQL> SELECT owner, table_name, grantor, privilege
        FROM user_tab_privs_recd ;
OWNER      TABLE_NAME      GRANTOR    PRIVILEGE
---------- --------------- ---------- ----------
ORA1       EMP             USER01     DELETE
SQL> CONN ora1/oracle
Connected.
SQL> REVOKE delete ON emp FROM user01 ;
Revoke succeeded.
SQL> CONN ora2/oracle
Connected.
SQL> SELECT owner, table_name, grantor, privilege
        FROM user_tab_privs_recd ;
no rows selected
SQL> CONN ora1/oracle
Connected.
```

4. 대형 데이터 집합 조작

- *Subquery 활용*

```
SQL> SELECT e.empno, e.ename, e.sal, e.deptno, a.avg
       FROM emp e JOIN ( SELECT deptno, AVG(sal) AS AVG
                              FROM emp
                              GROUP BY deptno ) a
          ON e.deptno = a.deptno AND e.sal > a.avg ;
     EMPNO ENAME             SAL     DEPTNO        AVG
---------- ---------- ---------- ---------- ----------
      7499 ALLEN            1600         30 1566.66667
      7566 JONES            2975         20       2175
      7698 BLAKE            2850         30 1566.66667
      7788 SCOTT            3000         20       2175
      7839 KING             5000         10 2916.66667
      7902 FORD             3000         20       2175
```

- *ROWNUM 사용*

```
SQL> SELECT ROWNUM, empno, ename, sal
       FROM emp ;
    ROWNUM      EMPNO ENAME             SAL
---------- ---------- ---------- ----------
         1       7369 SMITH             800
         2       7499 ALLEN            1600
         3       7521 WARD             1250
         4       7566 JONES            2975
         5       7654 MARTIN           1250
         6       7698 BLAKE            2850
         7       7782 CLARK            2450
         8       7788 SCOTT            3000
         9       7839 KING             5000
        10       7844 TURNER           1500
        11       7876 ADAMS            1100
        12       7900 JAMES             950
        13       7902 FORD             3000
        14       7934 MILLER           1300
SQL> SELECT ROWNUM, empno, ename, sal
       FROM emp
       ORDER BY sal DESC ;
    ROWNUM      EMPNO ENAME             SAL
---------- ---------- ---------- ----------
         9       7839 KING             5000
        13       7902 FORD             3000
         8       7788 SCOTT            3000
         4       7566 JONES            2975
         6       7698 BLAKE            2850
         7       7782 CLARK            2450
         2       7499 ALLEN            1600
        10       7844 TURNER           1500
        14       7934 MILLER           1300
         3       7521 WARD             1250
         5       7654 MARTIN           1250
        11       7876 ADAMS            1100
        12       7900 JAMES             950
         1       7369 SMITH             800
```

```
SQL> SELECT ROWNUM, empno, ename, sal
       FROM ( SELECT empno, ename, sal
                FROM emp
                ORDER BY sal DESC ) ;
    ROWNUM      EMPNO ENAME             SAL
---------- ---------- ---------- ----------
         1       7839 KING             5000
         2       7902 FORD             3000
         3       7788 SCOTT            3000
         4       7566 JONES            2975
         5       7698 BLAKE            2850
         6       7782 CLARK            2450
         7       7499 ALLEN            1600
         8       7844 TURNER           1500
         9       7934 MILLER           1300
        10       7521 WARD             1250
        11       7654 MARTIN           1250
        12       7876 ADAMS            1100
        13       7900 JAMES             950
        14       7369 SMITH             800
```

- *TOP-n 질의 사용*

```
SQL> SELECT ROWNUM, empno, ename, sal
       FROM ( SELECT empno, ename, sal
                FROM emp
                ORDER BY sal DESC )
       WHERE ROWNUM <= 3 ;
    ROWNUM      EMPNO ENAME             SAL
---------- ---------- ---------- ----------
         1       7839 KING             5000
         2       7788 SCOTT            3000
         3       7902 FORD             3000
```

- *Subquery 와 DML*

```
SQL> CREATE OR REPLACE VIEW empv10
       AS SELECT empno, ename, sal, deptno
          FROM emp
          WHERE deptno = 10 ;
View created.
SQL> UPDATE empv10
       SET sal = sal * 1.1
       WHERE empno = 7782 ;
1 row updated.
SQL> SELECT * FROM empv10 ;
     EMPNO ENAME             SAL     DEPTNO
---------- ---------- ---------- ----------
      7782 CLARK            2695         10
      7839 KING             5000         10
      7934 MILLER           1300         10
```

```
SQL> UPDATE (SELECT empno, ename, sal, deptno
                 FROM emp
                 WHERE deptno = 10)
        SET sal = sal * 1.1
        WHERE empno = 7782 ;
1 row updated.
SQL> SELECT empno, ename, sal, deptno
        FROM emp
        WHERE deptno = 10 ;
     EMPNO ENAME             SAL     DEPTNO
---------- ---------- ---------- ----------
      7782 CLARK          2964.5         10
      7839 KING             5000         10
      7934 MILLER           1300         10
SQL> ROLLBACK ;
Rollback complete.
SQL> CREATE OR REPLACE VIEW empv10
        AS SELECT empno, ename, sal, deptno
            FROM emp
            WHERE deptno = 10 WITH CHECK OPTION ;
View created.
SQL> UPDATE empv10
        SET deptno = 20
        WHERE empno = 7782 ;
ERROR at line 1:
ORA-01402: view WITH CHECK OPTION where-clause violation
SQL> UPDATE (SELECT empno, ename, sal, deptno
                 FROM emp
                 WHERE deptno = 10 WITH CHECK OPTION )
        SET deptno = 20
        WHERE empno = 7782 ;
ERROR at line 2:
ORA-01402: view WITH CHECK OPTION where-clause violation
```

※ 응용 사례

```
SQL> ALTER TABLE emp
        ADD ( dname      VARCHAR2(10) ) ;
Table altered.
SQL> SELECT empno, ename, deptno, dname
        FROM emp ;
     EMPNO ENAME          DEPTNO DNAME
---------- ---------- ---------- ----------
      7369 SMITH              20 (NULL)
      7499 ALLEN              30 (NULL)
      7521 WARD               30 (NULL)
      7566 JONES              20 (NULL)
      7654 MARTIN             30 (NULL)
      7698 BLAKE              30 (NULL)
...
```

```
SQL> UPDATE emp
       SET dname = 'ACCOUNTING'
       WHERE deptno = 10 ;
3 rows updated.
SQL> UPDATE emp
       SET dname = 'RESEARCH'
       WHERE deptno = 20 ;
5 rows updated.
SQL> UPDATE emp
       SET dname = 'SALES'
       WHERE deptno = 30 ;
6 rows updated.
SQL> SELECT empno, ename, deptno, dname
       FROM emp ;
     EMPNO ENAME          DEPTNO DNAME
---------- ---------- ---------- ----------
      7369 SMITH              20 RESEARCH
      7499 ALLEN              30 SALES
      7521 WARD               30 SALES
      7566 JONES              20 RESEARCH
      7654 MARTIN             30 SALES
...
SQL> ROLLBACK ;
Rollback complete.

SQL> SELECT e.deptno, e.dname, d.deptno, d.dname
       FROM emp e JOIN dept d
          ON e.deptno = d.deptno ;
    DEPTNO DNAME          DEPTNO DNAME
---------- ---------- ---------- --------------
        20 (NULL)             20 RESEARCH
        30 (NULL)             30 SALES
        30 (NULL)             30 SALES
        20 (NULL)             20 RESEARCH
        30 (NULL)             30 SALES
        30 (NULL)             30 SALES
        10 (NULL)             10 ACCOUNTING
...

SQL> ALTER TABLE dept
       ADD PRIMARY KEY (deptno) ;
Table altered.
SQL> ALTER TABLE emp
       ADD FOREIGN KEY (deptno) REFERENCES dept(deptno) ;
Table altered.
SQL> UPDATE   (SELECT e.deptno, e.dname emp_dname, d.deptno, d.dname dept_dname
                 FROM emp e JOIN dept d
                   ON e.deptno = d.deptno )
       SET emp_dname = dept_dname ;
14 rows updated.
```

```
SQL> SELECT empno, ename, deptno, dname
       FROM emp ;
EMPNO ENAME          DEPTNO DNAME
---------- ---------- ---------- ----------
      7369 SMITH             20 RESEARCH
      7499 ALLEN             30 SALES
      7521 WARD              30 SALES
      7566 JONES             20 RESEARCH
      7654 MARTIN            30 SALES
      7698 BLAKE             30 SALES
      7782 CLARK             10 ACCOUNTING
...
SQL> ROLLBACK ;
Rollback complete.

SQL> ALTER TABLE dept
       DROP PRIMARY KEY ;
ERROR at line 1:
ORA-02273: this unique/primary key is referenced by some foreign keys
SQL> ALTER TABLE dept
       DROP PRIMARY KEY CASCADE ;
Table altered.
```

※ Oracle Database 11g 부터는 bypass_ujvc 힌트 지원 안함

```
SQL> UPDATE   /*+ bypass_ujvc */
              (SELECT e.deptno, e.dname emp_dname, d.deptno, d.dname dept_dname
               FROM emp e JOIN dept d
               ON e.deptno = d.deptno )
       SET emp_dname = dept_dname ;
ERROR at line 5:
ORA-01779: cannot modify a column which maps to a non key-preserved table
```

※ Orcle Database 11g 부터는 제약조건이 없을 때 MERGE 사용

```
SQL> MERGE INTO emp e
       USING dept d
       ON (e.deptno = d.deptno)
       WHEN MATCHED THEN
       UPDATE
       SET e.dname = d.dname ;
14 rows merged.
SQL> SELECT empno, ename, deptno, dname
       FROM emp ;
EMPNO ENAME          DEPTNO DNAME
---------- ---------- ---------- ----------
      7369 SMITH             20 RESEARCH
      7499 ALLEN             30 SALES
      7521 WARD              30 SALES
      7566 JONES             20 RESEARCH
      7654 MARTIN            30 SALES
      7698 BLAKE             30 SALES
...
```

```
SQL> ALTER TABLE emp
       DROP COLUMN dname ;
Table altered.
```

- *다중 테이블의 INSERT*

```
SQL> CREATE TABLE sal_hist
       ( empno         NUMBER(4),
         ename         VARCHAR2(10),
         hiredate      DATE,
         sal           NUMBER(7,2)) ;
Table created.
SQL> CREATE TABLE mgr_hist
       ( empno         NUMBER(4),
         ename         VARCHAR2(10),
         hiredate      DATE,
         mgr           NUMBER(4)) ;
Table created.
SQL> INSERT INTO sal_hist
       SELECT empno, ename, hiredate, sal
       FROM emp
       WHERE hiredate < TO_DATE('82/12/31','RR/MM/DD') ;
13 rows created.
SQL> INSERT INTO mgr_hist
       SELECT empno, ename, hiredate, mgr
       FROM emp
       WHERE hiredate > TO_DATE('82/01/01','RR/MM/DD') ;
3 rows created.
SQL> SELECT * FROM sal_hist ;
EMPNO ENAME      HIREDATE         SAL
---------- ---------- --------- ----------
      7369 SMITH      17-DEC-80        800
      7499 ALLEN      20-FEB-81       1600
      7521 WARD       22-FEB-81       1250
      7566 JONES      02-APR-81       2975
      7654 MARTIN     28-SEP-81       1250
      7698 BLAKE      01-MAY-81       2850
      7782 CLARK      09-JUN-81       2450
      7788 SCOTT      09-DEC-82       3000
      7839 KING       17-NOV-81       5000
      7844 TURNER     08-SEP-81       1500
      7900 JAMES      03-DEC-81        950
      7902 FORD       03-DEC-81       3000
      7934 MILLER     23-JAN-82       1300
SQL> SELECT * FROM mgr_hist ;
EMPNO ENAME      HIREDATE         MGR
---------- ---------- --------- ----------
      7788 SCOTT      09-DEC-82       7566
      7876 ADAMS      12-JAN-83       7788
      7934 MILLER     23-JAN-82       7782
SQL> ROLLBACK ;
```

※ Unconditional INSERT ALL

```
SQL> INSERT ALL
        INTO sal_hist       VALUES (empno,ename,hiredate,sal)
        INTO mgr_hist       VALUES (empno,ename,hiredate,mgr)
       SELECT empno, ename, hiredate, sal, mgr
       FROM emp ;
28 rows created.
SQL> SELECT * FROM sal_hist ;
EMPNO ENAME      HIREDATE         SAL
---------- ---------- --------- ----------
      7369 SMITH      17-DEC-80        800
      7499 ALLEN      20-FEB-81       1600
      7521 WARD       22-FEB-81       1250
      7566 JONES      02-APR-81       2975
      7654 MARTIN     28-SEP-81       1250
      7698 BLAKE      01-MAY-81       2850
      7782 CLARK      09-JUN-81       2450
      7788 SCOTT      09-DEC-82       3000
      7839 KING       17-NOV-81       5000
      7844 TURNER     08-SEP-81       1500
      7876 ADAMS      12-JAN-83       1100
      7900 JAMES      03-DEC-81        950
      7902 FORD       03-DEC-81       3000
      7934 MILLER     23-JAN-82       1300
SQL> SELECT * FROM mgr_hist ;
EMPNO ENAME      HIREDATE         MGR
---------- ---------- --------- ----------
      7369 SMITH      17-DEC-80       7902
      7499 ALLEN      20-FEB-81       7698
      7521 WARD       22-FEB-81       7698
      7566 JONES      02-APR-81       7839
      7654 MARTIN     28-SEP-81       7698
      7698 BLAKE      01-MAY-81       7839
      7782 CLARK      09-JUN-81       7839
      7788 SCOTT      09-DEC-82       7566
      7839 KING       17-NOV-81 (NULL)
      7844 TURNER     08-SEP-81       7698
      7876 ADAMS      12-JAN-83       7788
      7900 JAMES      03-DEC-81       7698
      7902 FORD       03-DEC-81       7566
      7934 MILLER     23-JAN-82       7782
SQL> ROLLBACK ;
Rollback complete.
```

※ Conditional INSERT ALL

```
SQL> INSERT ALL
       WHEN hiredate < TO_DATE('82/12/31','RR/MM/DD') THEN
        INTO sal_hist       VALUES (empno,ename,hiredate,sal)
       WHEN hiredate > TO_DATE('82/01/01','RR/MM/DD') THEN
        INTO mgr_hist       VALUES (empno,ename,hiredate,mgr)
       SELECT empno, ename, hiredate, sal, mgr
       FROM emp ;
16 rows created.
```

```
SQL> SELECT * FROM sal_hist ;
     EMPNO ENAME      HIREDATE         SAL
---------- ---------- --------- ----------
      7369 SMITH      17-DEC-80        800
      7499 ALLEN      20-FEB-81       1600
      7521 WARD       22-FEB-81       1250
      7566 JONES      02-APR-81       2975
      7654 MARTIN     28-SEP-81       1250
      7698 BLAKE      01-MAY-81       2850
      7782 CLARK      09-JUN-81       2450
      7788 SCOTT      09-DEC-82       3000
      7839 KING       17-NOV-81       5000
      7844 TURNER     08-SEP-81       1500
      7900 JAMES      03-DEC-81        950
      7902 FORD       03-DEC-81       3000
      7934 MILLER     23-JAN-82       1300
SQL> SELECT * FROM mgr_hist ;
     EMPNO ENAME      HIREDATE         MGR
---------- ---------- --------- ----------
      7788 SCOTT      09-DEC-82       7566
      7876 ADAMS      12-JAN-83       7788
      7934 MILLER     23-JAN-82       7782
SQL> ROLLBACK ;
Rollback complete.
```

※ Conditional INSERT FIRST

```
SQL> INSERT FIRST
       WHEN hiredate < TO_DATE('82/12/31','RR/MM/DD') THEN
        INTO sal_hist      VALUES (empno,ename,hiredate,sal)
       WHEN hiredate > TO_DATE('82/01/01','RR/MM/DD') THEN
        INTO mgr_hist      VALUES (empno,ename,hiredate,mgr)
       SELECT empno, ename, hiredate, sal, mgr
       FROM emp ;
14 rows created.
SQL> SELECT * FROM sal_hist ;
     EMPNO ENAME      HIREDATE         SAL
---------- ---------- --------- ----------
      7369 SMITH      17-DEC-80        800
      7499 ALLEN      20-FEB-81       1600
      7521 WARD       22-FEB-81       1250
      7566 JONES      02-APR-81       2975
      7654 MARTIN     28-SEP-81       1250
      7698 BLAKE      01-MAY-81       2850
      7782 CLARK      09-JUN-81       2450
      7788 SCOTT      09-DEC-82       3000
      7839 KING       17-NOV-81       5000
      7844 TURNER     08-SEP-81       1500
      7900 JAMES      03-DEC-81        950
      7902 FORD       03-DEC-81       3000
      7934 MILLER     23-JAN-82       1300
SQL> SELECT * FROM mgr_hist ;
EMPNO ENAME      HIREDATE         MGR
---------- ---------- --------- ----------
      7876 ADAMS      12-JAN-83       7788
SQL> ROLLBACK ;
Rollback complete.
```

※ Pivot INSERT ALL

```
SQL> CREATE TABLE source_data
       ( empno              NUMBER(4),
         week_id            NUMBER(2),
         sales_mon          NUMBER(4),
         sales_tue          NUMBER(4),
         sales_wed          NUMBER(4),
         sales_thur         NUMBER(4),
         sales_fri          NUMBER(4) ) ;
Table created.
SQL> INSERT INTO source_data
       VALUES (100, 1, 1000, 1200, 500, 600, 1300 ) ;
1 row created.
SQL> INSERT INTO source_data
       VALUES (101, 1, 1200, 1300, 1500, 600, 1000 ) ;
1 row created.
SQL> SELECT * FROM source_data ;
     EMPNO    WEEK_ID  SALES_MON  SALES_TUE  SALES_WED SALES_THUR  SALES_FRI
---------- ---------- ---------- ---------- ---------- ---------- ----------
       100          1       1000       1200        500        600       1300
       101          1       1200       1300       1500        600       1000
SQL> SELECT SUM(sales_mon), SUM(sales_tue), SUM(sales_wed), SUM(sales_thur), SUM(sales_fri)
       FROM source_data ;
SUM(SALES_MON) SUM(SALES_TUE) SUM(SALES_WED) SUM(SALES_THUR) SUM(SALES_FRI)
-------------- -------------- -------------- --------------- --------------
          2200           2500           2000            1200           2300
SQL> CREATE TABLE sales_info
       ( empno              NUMBER(4),
         week_id            NUMBER(2),
         day                VARCHAR(4),
         sales              NUMBER(4) ) ;
Table created.
SQL> INSERT ALL
         INTO sales_info VALUES (empno, week_id, 'MON', sales_mon)
         INTO sales_info VALUES (empno, week_id, 'TUE', sales_tue)
         INTO sales_info VALUES (empno, week_id, 'WED', sales_wed)
         INTO sales_info VALUES (empno, week_id, 'THUR', sales_thur)
         INTO sales_info VALUES (empno, week_id, 'FRI', sales_fri)
       SELECT * FROM source_data ;
10 rows created.
```

```
SQL> SELECT * FROM sales_info ;
     EMPNO    WEEK_ID DAY       SALES
---------- ---------- ---- ----------
       100          1 MON        1000
       101          1 MON        1200
       100          1 TUE        1200
       101          1 TUE        1300
       100          1 WED         500
       101          1 WED        1500
       100          1 THUR        600
       101          1 THUR        600
       100          1 FRI        1300
       101          1 FRI        1000
SQL> SELECT day, SUM(sales)
       FROM sales_info
       GROUP BY day ;
DAY  SUM(SALES)
---- ----------
TUE        2500
THUR       1200
FRI        2300
MON        2200
WED        2000
```

※ PIVOT 사용

```
SQL> SELECT *
       FROM ( SELECT day, sales FROM sales_info )
       PIVOT ( SUM(sales) FOR day IN ('MON' AS SALES_MON,
                                      'TUE'   AS SALES_TUE,
                                      'WED' AS SALES_WED,
                                      'THUR' AS SALES_THUR,
                                      'FRI'   AS SALES_FRI)) ;
SALES_MON  SALES_TUE  SALES_WED SALES_THUR  SALES_FRI
---------- ---------- ---------- ---------- ----------
      2200       2500       2000       1200       2300
```

※ UNPIVOT 사용

```
SQL> SELECT empno, week_id, SUBSTR(day,7) AS day, sales
       FROM source_data
       UNPIVOT ( sales FOR day IN (SALES_MON,SALES_TUE,SALES_WED,SALES_THUR,SALES_FRI)) ;
EMPNO    WEEK_ID DAY                   SALES
---------- ---------- ---------------- ----------
       100          1 MON                    1000
       100          1 TUE                    1200
       100          1 WED                     500
       100          1 THUR                    600
       100          1 FRI                    1300
       101          1 MON                    1200
       101          1 TUE                    1300
       101          1 WED                    1500
       101          1 THUR                    600
       101          1 FRI                    1000
```

```
SQL> DROP TABLE sal_hist PURGE ;
Table dropped.
SQL> DROP TABLE mgr_hist PURGE ;
Table dropped.
SQL> DROP TABLE source_data PURGE ;
Table dropped.
SQL> DROP TABLE sales_info PURGE ;
Table dropped.
```

- *MERGE 문 사용*

```
SQL> DROP TABLE copy_emp PURGE ;
Table dropped.
SQL> CREATE TABLE copy_emp
       AS
       SELECT * FROM emp
       WHERE deptno = 30 ;
Table created.
SQL> UPDATE copy_emp
       SET sal = sal * 0.5
       WHERE job = 'SALESMAN' ;
4 rows updated.
SQL> COMMIT ;
Commit complete.
SQL> SELECT ename, sal, comm, deptno
       FROM emp
       ORDER BY deptno, sal ;
ENAME             SAL       COMM     DEPTNO
---------- ---------- ---------- ----------
MILLER           1300 (NULL)             10
CLARK            2450 (NULL)             10
KING             5000 (NULL)             10
SMITH             800 (NULL)             20
ADAMS            1100 (NULL)             20
JONES            2975 (NULL)             20
SCOTT            3000 (NULL)             20
FORD             3000 (NULL)             20
JAMES             950 (NULL)             30
MARTIN           1250       1400         30
WARD             1250        500         30
TURNER           1500          0         30
ALLEN            1600        300         30
BLAKE            2850 (NULL)             30
SQL> SELECT ename, sal, comm, deptno
       FROM copy_emp
       ORDER BY deptno, sal ;
ENAME             SAL       COMM     DEPTNO
---------- ---------- ---------- ----------
MARTIN            625       1400         30
WARD              625        500         30
TURNER            750          0         30
ALLEN             800        300         30
JAMES             950 (NULL)             30
BLAKE            2850 (NULL)             30
```

```
SQL> MERGE INTO copy_emp c
       USING emp e
       ON (c.empno = e.empno)
       WHEN MATCHED THEN
       UPDATE
       SET c.sal     = e.sal ,
           c.comm = e.comm
       WHEN NOT MATCHED THEN
       INSERT
       VALUES (e.empno,e.ename,e.job, e.mgr, e.hiredate, e.sal, e.comm, e.deptno) ;
14 rows merged.
SQL> SELECT ename, sal, comm, deptno
       FROM copy_emp
       ORDER BY deptno, sal ;
ENAME             SAL       COMM     DEPTNO
---------- ---------- ---------- ----------
MILLER           1300 (NULL)             10
CLARK            2450 (NULL)             10
KING             5000 (NULL)             10
SMITH             800 (NULL)             20
ADAMS            1100 (NULL)             20
JONES            2975 (NULL)             20
SCOTT            3000 (NULL)             20
FORD             3000 (NULL)             20
JAMES             950 (NULL)             30
MARTIN           1250       1400         30
WARD             1250        500         30
TURNER           1500          0         30
ALLEN            1600        300         30
BLAKE            2850 (NULL)             30
SQL> ROLLBACK ;
Rollback complete.

SQL> MERGE INTO copy_emp c
       USING emp e
       ON (c.empno = e.empno)
       WHEN MATCHED THEN
       UPDATE
       SET c.sal     = e.sal ,
           c.comm = e.comm
       DELETE WHERE (c.comm IS NOT NULL)
       WHEN NOT MATCHED THEN
       INSERT
       VALUES (e.empno,e.ename,e.job, e.mgr, e.hiredate, e.sal, e.comm, e.deptno) ;
14 rows merged.
```

```
SQL> SELECT ename, sal, comm, deptno
     FROM copy_emp
     ORDER BY deptno, sal ;
ENAME             SAL       COMM     DEPTNO
---------- ---------- ---------- ----------
MILLER           1300 (NULL)             10
CLARK            2450 (NULL)             10
KING             5000 (NULL)             10
SMITH             800 (NULL)             20
ADAMS            1100 (NULL)             20
JONES            2975 (NULL)             20
SCOTT            3000 (NULL)             20
FORD             3000 (NULL)             20
JAMES             950 (NULL)             30
BLAKE            2850 (NULL)             30

SQL> DROP TABLE copy_emp PURGE ;
Table dropped.
```

6. Subquery 를 사용하여 데이터 검색

- *Multiple-Column Subquery 사용*

```
SQL> SELECT empno, ename, sal, deptno
       FROM emp
       WHERE (deptno, sal) IN ( SELECT deptno, MIN(sal)
                                FROM emp
                                GROUP BY deptno ) ;
     EMPNO ENAME             SAL     DEPTNO
---------- ---------- ---------- ----------
      7900 JAMES             950         30
      7369 SMITH             800         20
      7934 MILLER           1300         10
SQL> SELECT empno, ename, sal, deptno
       FROM emp
       WHERE deptno IN ( SELECT deptno FROM emp )
         AND sal      IN ( SELECT MIN(sal) FROM emp
                          GROUP BY deptno ) ;
     EMPNO ENAME             SAL     DEPTNO
---------- ---------- ---------- ----------
      7369 SMITH             800         20
      7900 JAMES             950         30
      7934 MILLER           1300         10

SQL> UPDATE emp
       SET sal = 950
       WHERE empno = 7876 ;
1 row updated.
SQL> SELECT empno, ename, sal, deptno
       FROM emp
       WHERE (deptno, sal) IN ( SELECT deptno, MIN(sal)
                                FROM emp
                                GROUP BY deptno ) ;
     EMPNO ENAME             SAL     DEPTNO
---------- ---------- ---------- ----------
      7900 JAMES             950         30
      7369 SMITH             800         20
      7934 MILLER           1300         10
SQL> SELECT empno, ename, sal, deptno
       FROM emp
       WHERE deptno IN ( SELECT deptno FROM emp )
         AND sal      IN ( SELECT MIN(sal) FROM emp
                          GROUP BY deptno ) ;
     EMPNO ENAME             SAL     DEPTNO
---------- ---------- ---------- ----------
      7876 ADAMS             950         20
      7369 SMITH             800         20
      7900 JAMES             950         30
      7934 MILLER           1300         10
```

```
SQL> ROLLBACK ;
Rollback complete.
```

- *Scalar Subquery (Correlated Subquery) 사용*

```
SQL> SELECT e.ename, e.sal, e.deptno
       FROM emp e JOIN ( SELECT deptno, AVG(sal) AS AVG
                           FROM emp
                           GROUP BY deptno ) a
          ON e.deptno = a.deptno
         AND e.sal > a.avg ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
ALLEN            1600         30
JONES            2975         20
BLAKE            2850         30
SCOTT            3000         20
KING             5000         10
FORD             3000         20
SQL> SELECT ename, sal, deptno
       FROM emp e
       WHERE sal > ( SELECT AVG(sal)
                       FROM emp
                       WHERE deptno = e.deptno ) ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
ALLEN            1600         30
JONES            2975         20
BLAKE            2850         30
SCOTT            3000         20
KING             5000         10
FORD             3000         20
```

※ 분석 함수 사용

```
SQL> SELECT ename, sal, deptno
       FROM ( SELECT ename, sal, deptno, AVG(sal) OVER(PARTITION BY deptno) AS AVG_OVER
                FROM emp )
       WHERE sal > avg_over ;
ENAME             SAL     DEPTNO
---------- ---------- ----------
KING             5000         10
JONES            2975         20
FORD             3000         20
SCOTT            3000         20
ALLEN            1600         30
BLAKE            2850         30
```

※ EXISTS 사용

```
SQL> SELECT empno, ename, sal, deptno
       FROM emp
       WHERE empno IN ( SELECT mgr FROM emp ) ;
     EMPNO ENAME             SAL     DEPTNO
---------- ---------- ---------- ----------
      7902 FORD             3000         20
      7698 BLAKE            2850         30
      7839 KING             5000         10
      7566 JONES            2975         20
      7788 SCOTT            3000         20
      7782 CLARK            2450         10
SQL> SELECT empno, ename, sal, deptno
       FROM emp e
       WHERE EXISTS ( SELECT 1
                       FROM emp
                       WHERE mgr = e.empno ) ;
     EMPNO ENAME             SAL     DEPTNO
---------- ---------- ---------- ----------
      7902 FORD             3000         20
      7698 BLAKE            2850         30
      7839 KING             5000         10
      7566 JONES            2975         20
      7788 SCOTT            3000         20
      7782 CLARK            2450         10
```

```
SQL> SELECT empno, ename, sal, deptno
       FROM emp
       WHERE empno NOT IN ( SELECT mgr FROM emp ) ;
no rows selected
SQL> SELECT empno, ename, sal, deptno
       FROM emp e
       WHERE NOT EXISTS ( SELECT 1
                           FROM emp
                           WHERE mgr = e.empno ) ;
     EMPNO ENAME             SAL     DEPTNO
---------- ---------- ---------- ----------
      7844 TURNER           1500         30
      7521 WARD             1250         30
      7654 MARTIN           1250         30
      7499 ALLEN            1600         30
      7934 MILLER           1300         10
      7369 SMITH             800         20
      7876 ADAMS            1100         20
      7900 JAMES             950         30
```

- *WITH 절 사용*

```
SQL> SELECT deptno, SUM(sal)
       FROM emp
       GROUP BY deptno
       HAVING SUM(sal) > ( SELECT AVG(SUM(sal))
                           FROM emp
                           GROUP BY deptno ) ;
    DEPTNO   SUM(SAL)
---------- ----------
        20      10875
SQL> WITH emp_sum AS ( SELECT deptno, SUM(sal) AS SUM
                       FROM emp
                       GROUP BY deptno )
       SELECT *
       FROM emp_sum
       WHERE sum > ( SELECT AVG(sum) FROM emp_sum ) ;
    DEPTNO        SUM
---------- ----------
        20      10875

SQL> SET AUTOTRACE ON EXPLAIN
SQL> SELECT deptno, SUM(sal)
       FROM emp
       GROUP BY deptno
       HAVING SUM(sal) > ( SELECT AVG(SUM(sal))
                           FROM emp
                           GROUP BY deptno ) ;
Execution Plan
----------------------------------------------------------
Plan hash value: 1944742949
-----------------------------------------------------------------------------
| Id  | Operation            | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------
|   0 | SELECT STATEMENT     |      |     1 |     7 |     4  (25)| 00:00:01 |
|*  1 |  FILTER              |      |       |       |            |          |
|   2 |   HASH GROUP BY      |      |     1 |     7 |     4  (25)| 00:00:01 |
|   3 |    TABLE ACCESS FULL | EMP  |    14 |    98 |     3   (0)| 00:00:01 |
|   4 |   SORT AGGREGATE     |      |     1 |     7 |     4  (25)| 00:00:01 |
|   5 |    SORT GROUP BY     |      |     1 |     7 |     4  (25)| 00:00:01 |
|   6 |     TABLE ACCESS FULL| EMP  |    14 |    98 |     3   (0)| 00:00:01 |
-----------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------
   1 - filter(SUM("SAL")> (SELECT AVG(SUM("SAL")) FROM "EMP" "EMP"
              GROUP BY "DEPTNO"))
```

```
SQL> WITH emp_sum AS ( SELECT deptno, SUM(sal) AS SUM
                       FROM emp
                       GROUP BY deptno )
       SELECT *
       FROM emp_sum
       WHERE sum > ( SELECT AVG(sum) FROM emp_sum ) ;
Execution Plan
----------------------------------------------------------
Plan hash value: 1063747289
--------------------------------------------------------------------------------------------------------
| Id  | Operation                  | Name                        | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT           |                             |     3 |    78 |     8  (13)| 00:00:01 |
|   1 |  TEMP TABLE TRANSFORMATION |                             |       |       |            |          |
|   2 |   LOAD AS SELECT           | SYS_TEMP_0FD9D6612_124BF8 |       |       |            |          |
|   3 |    HASH GROUP BY           |                             |     3 |    21 |     4  (25)| 00:00:01 |
|   4 |     TABLE ACCESS FULL      | EMP                         |    14 |    98 |     3   (0)| 00:00:01 |
|*  5 |   VIEW                     |                             |     3 |    78 |     2   (0)| 00:00:01 |
|   6 |    TABLE ACCESS FULL       | SYS_TEMP_0FD9D6612_124BF8 |     3 |    21 |     2   (0)| 00:00:01 |
|   7 |    SORT AGGREGATE          |                             |     1 |    13 |            |          |
|   8 |     VIEW                   |                             |     3 |    39 |     2   (0)| 00:00:01 |
|   9 |      TABLE ACCESS FULL     | SYS_TEMP_0FD9D6612_124BF8 |     3 |    21 |     2   (0)| 00:00:01 |
--------------------------------------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------
   5 - filter("SUM"> (SELECT AVG("SUM") FROM  (SELECT /*+ CACHE_TEMP_TABLE ("T1") */ "C0"
              "DEPTNO","C1" "SUM" FROM "SYS"."SYS_TEMP_0FD9D6612_124BF8" "T1") "EMP_SUM"))

SQL> SELECT deptno, sum
       FROM (SELECT deptno, sum, AVG(sum) OVER() AVG_OVER
             FROM ( SELECT deptno, SUM(sal) SUM
                    FROM emp
                    GROUP BY deptno ))
       WHERE sum > avg_over ;
DEPTNO        SUM
---------- ----------
        20      10875
Execution Plan
----------------------------------------------------------
Plan hash value: 4098264197
-----------------------------------------------------------------------------
| Id  | Operation            | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------
|   0 | SELECT STATEMENT     |      |     3 |   117 |     4  (25)| 00:00:01 |
|*  1 |  VIEW                |      |     3 |   117 |     4  (25)| 00:00:01 |
|   2 |   WINDOW BUFFER      |      |     3 |    21 |     4  (25)| 00:00:01 |
|   3 |    HASH GROUP BY     |      |     3 |    21 |     4  (25)| 00:00:01 |
|   4 |     TABLE ACCESS FULL| EMP  |    14 |    98 |     3   (0)| 00:00:01 |
-----------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------
   1 - filter("SUM">"AVG_OVER")

SQL> SET AUTOTRACE OFF
```

• Recursive WITH 절 사용

※ 계층질의 사용

```
SQL> SELECT LEVEL, ename, empno, mgr
       FROM emp
       START WITH mgr IS NULL
       CONNECT BY PRIOR empno = mgr ;
     LEVEL ENAME           EMPNO        MGR
---------- ---------- ---------- ----------
         1 KING             7839 (NULL)
         2 JONES            7566       7839
         3 SCOTT            7788       7566
         4 ADAMS            7876       7788
         3 FORD             7902       7566
         4 SMITH            7369       7902
         2 BLAKE            7698       7839
         3 ALLEN            7499       7698
         3 WARD             7521       7698
         3 MARTIN           7654       7698
         3 TURNER           7844       7698
         3 JAMES            7900       7698
         2 CLARK            7782       7839
         3 MILLER           7934       7782
```

```
SQL> SELECT LPAD(' ',level*2-2)||ename AS NAME, LEVEL, empno, mgr
       FROM emp
       START WITH mgr IS NULL
       CONNECT BY PRIOR empno = mgr ;
NAME                      LEVEL      EMPNO        MGR
--------------- ---------- ---------- ----------
KING                          1       7839 (NULL)
  JONES                       2       7566       7839
    SCOTT                     3       7788       7566
      ADAMS                   4       7876       7788
    FORD                      3       7902       7566
      SMITH                   4       7369       7902
  BLAKE                       2       7698       7839
    ALLEN                     3       7499       7698
    WARD                      3       7521       7698
    MARTIN                    3       7654       7698
    TURNER                    3       7844       7698
    JAMES                     3       7900       7698
  CLARK                       2       7782       7839
    MILLER                    3       7934       7782
```

※ Recursive WITH 절 사용

```
SQL> WITH htree (hlevel, ename, empno, mgr)
       AS ( SELECT 1 AS hlevel , ename, empno, mgr
            FROM emp
            WHERE mgr IS NULL
            UNION ALL
            SELECT hlevel + 1, e.ename, e.empno, e.mgr
            FROM emp e, htree h
            WHERE e.mgr = h.empno )
       SELECT * FROM htree ;
    HLEVEL ENAME           EMPNO        MGR
---------- ---------- ---------- ----------
         1 KING             7839 (NULL)
         2 JONES            7566       7839
         2 BLAKE            7698       7839
         2 CLARK            7782       7839
         3 ALLEN            7499       7698
         3 WARD             7521       7698
         3 MARTIN           7654       7698
         3 SCOTT            7788       7566
         3 TURNER           7844       7698
         3 JAMES            7900       7698
         3 FORD             7902       7566
         3 MILLER           7934       7782
         4 SMITH            7369       7902
         4 ADAMS            7876       7788
```

```
SQL> WITH htree (hlevel, ename, empno, mgr)
       AS ( SELECT 1 AS hlevel , ename, empno, mgr
            FROM emp
            WHERE mgr IS NULL
            UNION ALL
            SELECT hlevel + 1, e.ename, e.empno, e.mgr
            FROM emp e, htree h
            WHERE e.mgr = h.empno )
       SELECT LPAD(' ',hlevel*2-2)||ename AS NAME, hlevel, empno, mgr
       FROM htree ;
NAME                HLEVEL      EMPNO        MGR
--------------- ---------- ---------- ----------
KING                     1       7839 (NULL)
  JONES                  2       7566       7839
  BLAKE                  2       7698       7839
  CLARK                  2       7782       7839
    ALLEN                3       7499       7698
    WARD                 3       7521       7698
    MARTIN               3       7654       7698
    SCOTT                3       7788       7566
    TURNER               3       7844       7698
    JAMES                3       7900       7698
    FORD                 3       7902       7566
    MILLER               3       7934       7782
      SMITH              4       7369       7902
      ADAMS              4       7876       7788
```

```
SQL> WITH htree (hlevel, ename, empno, mgr)
       AS ( SELECT 1 AS hlevel , ename, empno, mgr
            FROM emp
            WHERE mgr IS NULL
            UNION ALL
            SELECT hlevel + 1, e.ename, e.empno, e.mgr
            FROM emp e, htree h
            WHERE e.mgr = h.empno )
       SEARCH DEPTH FIRST BY empno SET IDX
       SELECT LPAD(' ',hlevel*2-2)||ename AS NAME, hlevel, empno, mgr
       FROM htree ;
NAME                HLEVEL      EMPNO        MGR
--------------- ---------- ---------- ----------
KING                     1       7839 (NULL)
  JONES                  2       7566       7839
    SCOTT                3       7788       7566
      ADAMS              4       7876       7788
    FORD                 3       7902       7566
      SMITH              4       7369       7902
  BLAKE                  2       7698       7839
    ALLEN                3       7499       7698
    WARD                 3       7521       7698
    MARTIN               3       7654       7698
    TURNER               3       7844       7698
    JAMES                3       7900       7698
  CLARK                  2       7782       7839
    MILLER               3       7934       7782
```

```
SQL> CREATE TABLE flights
       ( source       VARCHAR2(20),
         destin       VARCHAR2(20),
         flight_time  NUMBER(2,1) ) ;
Table created.
SQL> INSERT INTO flights
       VALUES ('San Jose','Los Angeles',1.3) ;
1 row created.
SQL> INSERT INTO flights
       VALUES ('New York','Boston',1.1) ;
1 row created.
SQL> INSERT INTO flights
       VALUES ('Los Angeles','New York',5.8) ;
1 row created.
SQ> COMMIT ;
Commit complete.
```

```
SQL> SELECT * FROM flights;
SOURCE               DESTIN               FLIGHT_TIME
-------------------- -------------------- -----------
San Jose             Los Angeles                  1.3
New York             Boston                       1.1
Los Angeles          New York                     5.8
```

```
SQL> WITH RECUR ( source, destin, flighttime ) AS
        ( SELECT source, destin, flight_time
          FROM flights
          UNION ALL
          SELECT incoming.source, outgoing.destin, incoming.flighttime + outgoing.flight_time
          FROM recur incoming, flights outgoing
          WHERE incoming.destin = outgoing.source )
        SELECT source, destin, flighttime
        FROM recur ;
SOURCE               DESTIN               FLIGHTTIME
-------------------- -------------------- ----------
San Jose             Los Angeles                 1.3
New York             Boston                      1.1
Los Angeles          New York                    5.8
Los Angeles          Boston                      6.9
San Jose             New York                    7.1
San Jose             Boston                      8.2

SQL> SELECT CASE level WHEN 1 THEN source ELSE CONNECT_BY_ROOT source END AS source,
                destin,
                SUM(flight_time) OVER(PARTITION BY CONNECT_BY_ROOT SOURCE
                                      ORDER BY LEVEL) flighttime
        FROM flights
        CONNECT BY source = PRIOR destin ;
SOURCE               DESTIN               FLIGHTTIME
-------------------- -------------------- ----------
Los Angeles          New York                    5.8
Los Angeles          Boston                      6.9
New York             Boston                      1.1
San Jose             Los Angeles                 1.3
San Jose             New York                    7.1
San Jose             Boston                      8.2

SQL> exit
```