

이 장에서 다룰 내용



- 1 **서브 쿼리의 기본 개념**
- 2 **단일 행 서브 쿼리**
- 3 **서브 쿼리에서 그룹 함수의 사용**
- 4 **다중 행 서브 쿼리**

01. 서브 쿼리의 기본 개념

- SCOTT의 부서명을 알아봅시다.



- ❖ 1. SCOTT의 부서번호를 알아내기 위한 쿼리문 작성해주세요

```
SELECT DEPTNO  
FROM EMP  
WHERE ENAME='SCOTT'
```

- ❖ 2. 1번에서 구한 부서번호에 해당하는 부서명을 구해보세요.

```
SELECT DNAME  
FROM DEPT  
WHERE DEPTNO = 20;
```

01. 서브 쿼리의 기본 개념



- ❖ SCOTT의 부서명을 알아내기 위한 서브 쿼리문 부터 살펴봅시다.

메인 쿼리

```
SELECT DNAME  
FROM DEPT
```

```
WHERE DEPTNO = ( SELECT DEPTNO  
                  FROM EMP  
                  WHERE ENAME='SCOTT' )
```

서브 쿼리

01. 서브 쿼리의 기본 개념



- ❖ 서브 쿼리는 하나의 SQL 문장의 절 안에 포함된 또 하나의 SELECT 문장입니다.
- ❖ 그렇기에 서브 쿼리를 포함하고 있는 쿼리문을 메인 쿼리, 포함된 또 하나의 쿼리를 서브 쿼리라 합니다.
- ❖ 서브 쿼리는 비교 연산자의 오른쪽에 기술해야 하고 반드시 괄호로 둘러싸아야 합니다.
- ❖ 서브 쿼리는 메인 쿼리가 실행되기 이전에 한번만 실행이 됩니다.
- ❖ 서브쿼리의 위치
 - 1) SELECT , DELETE의 FROM절, WHERE 절
 - 2) INSERT 문장의 VALUES절
 - 3) UPDATE 문장의 SET절, WHERE절

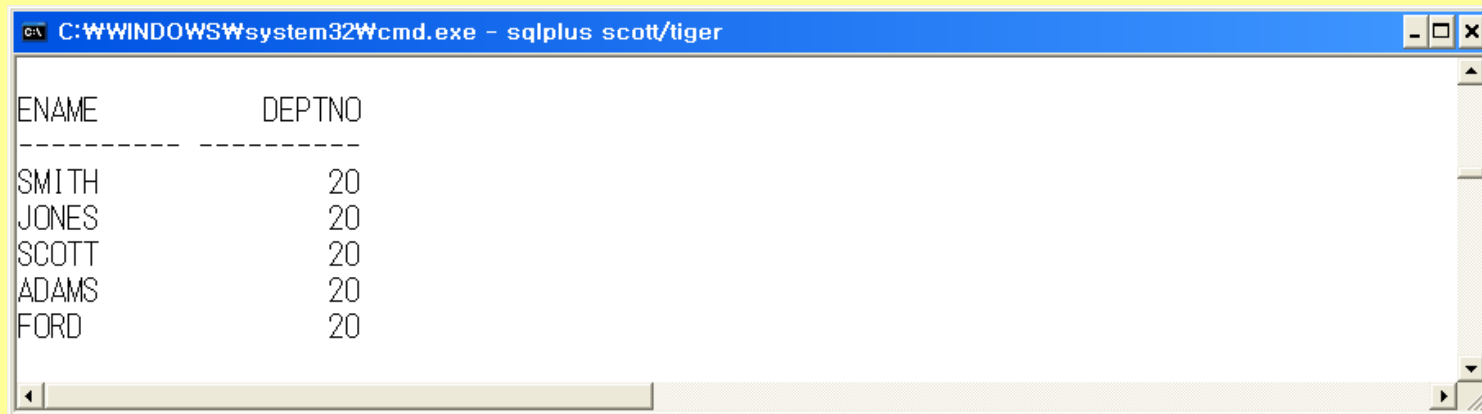
02. 단일 행 서브 쿼리



- ❖ 단일 행(Single Row) 서브 쿼리는 수행 결과가 오직 하나의 로우(행, row)만을 반환하는 서브 쿼리를 갖는 것을 말합니다.
- ❖ 단일 행 서브 쿼리문에서는 이렇게 오직 하나의 로우(행, row)로 반환되는 서브 쿼리의 결과는 메인 쿼리에 보내게 되는데 메인 쿼리의 WHERE 절에서는 단일 행 비교 연산자인 =, >, >=, <, <=, <>를 사용해야 합니다.

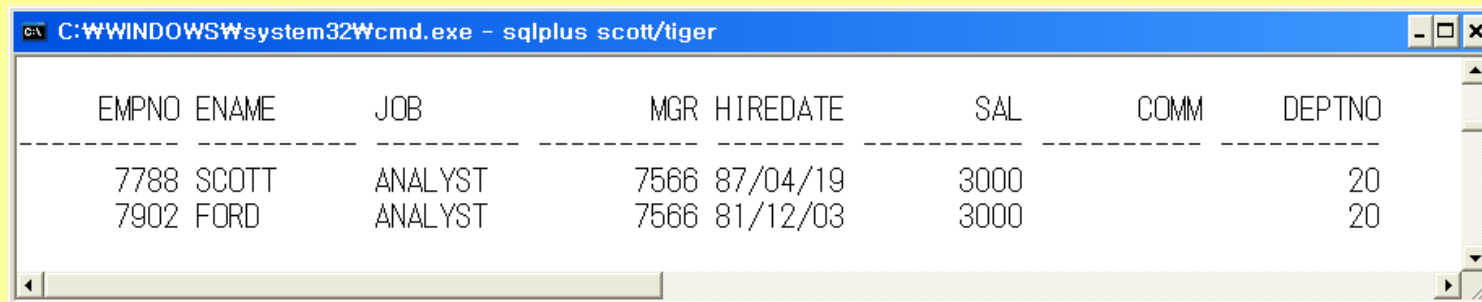
<탄탄히 다지기>

1. SCOTT과 같은 부서에서 근무하는 사원의 이름과 부서 번호를 출력하는 SQL 문을 작성해 보시오.



ENAME	DEPTNO
SMITH	20
JONES	20
SCOTT	20
ADAMS	20
FORD	20

2. SCOTT와 동일한 업무(JOB)를 가진 사원을 출력하는 SQL 문을 작성해 보시오.



EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7788	SCOTT	ANALYST	7566	87/04/19	3000		20
7902	FORD	ANALYST	7566	81/12/03	3000		20

<탄탄히 다지기>

3. SCOTT의 급여와 동일하거나 더 많이 받는 사원명과 급여를 출력하시오.

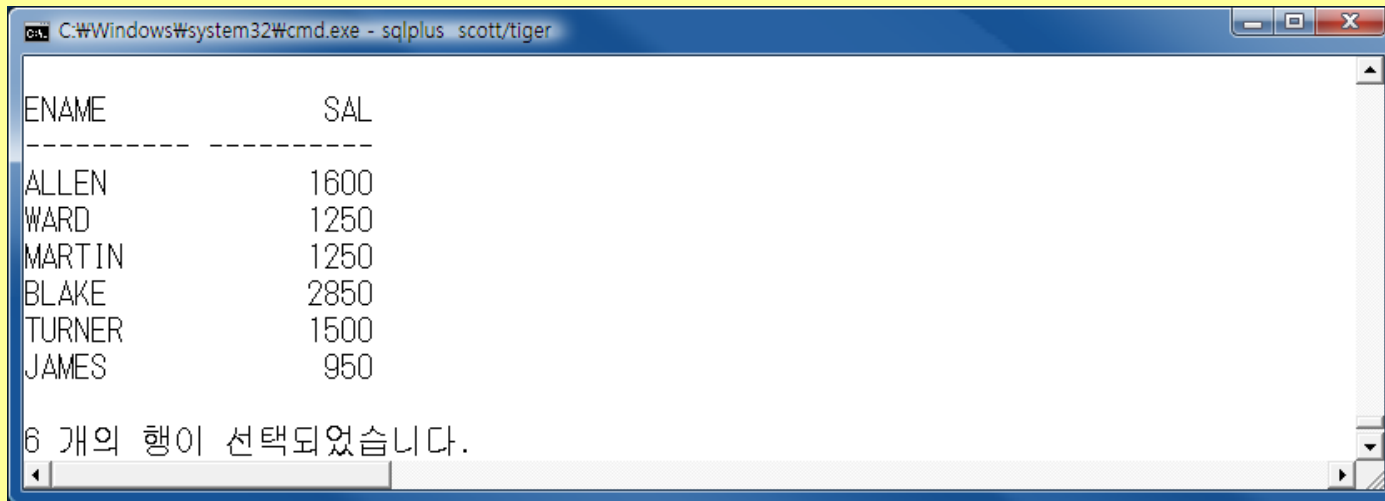
ENAME	SAL
SCOTT	3000
KING	5000
FORD	3000

4. DALLAS에서 근무하는 사원의 이름, 부서 번호를 출력하시오.

ENAME	DEPTNO
SMITH	20
JONES	20
SCOTT	20
ADAMS	20
FORD	20

<탄탄히 다지기>

5. SALES(영업부) 부서에서 근무하는 모든 사원의 이름과 급여를 출력하시오.

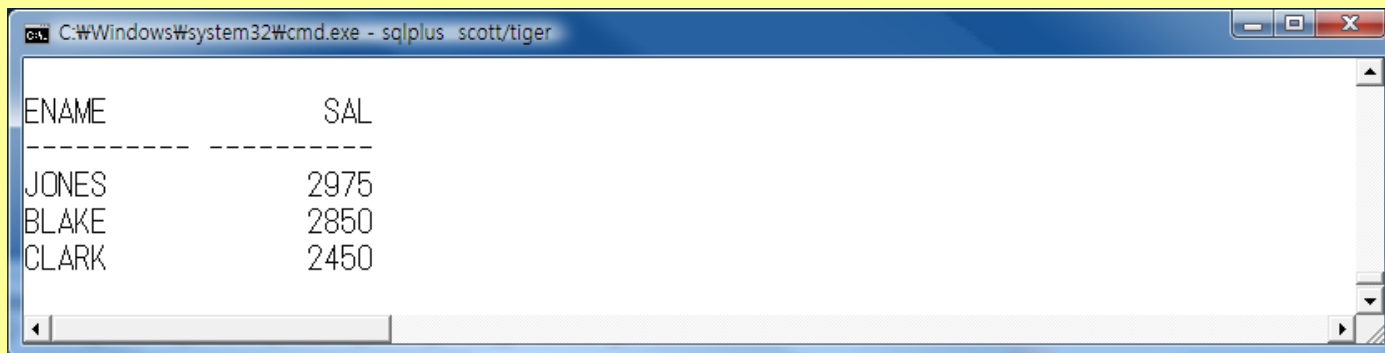


```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger
```

ENAME	SAL
ALLEN	1600
WARD	1250
MARTIN	1250
BLAKE	2850
TURNER	1500
JAMES	950

6 개의 행이 선택되었습니다.

6. 직속상관이 KING인 사원의 이름과 급여를 출력하시오.



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

ENAME	SAL
JONES	2975
BLAKE	2850
CLARK	2450

03. 서브 쿼리에서 그룹 함수의 사용



- ❖ 평균 급여를 구하는 쿼리문을 서브 쿼리로 사용하여 평균 급여보다 더 많은 급여를 받는 사원을 검색하는 문장은 다음과 같습니다.

```
SELECT ENAME, SAL
FROM EMP
WHERE SAL > ( SELECT AVG(SAL)
               FROM   EMP);
```

ENAME	SAL
JONES	2975
BLAKE	2850
CLARK	2450
SCOTT	3000
KING	5000
FORD	3000

6 rows selected.

04. 다중 행 서브 쿼리



- ❖ 다중 행 서브 쿼리는 서브 쿼리에서 반환되는 결과가 하나 이상의 행일 때 사용하는 서브 쿼리입니다. 다중 행 서브 쿼리는 반드시 다중 행 연산자(Multiple Row Operator)와 함께 사용해야 합니다.

종류	의미
IN	메인 쿼리의 비교 조건('=' 연산자로 비교할 경우)이 서브 쿼리의 결과 중에서 하나라도 일치하면 참입니다.
ANY	메인 쿼리의 비교 조건이 서브 쿼리의 검색 결과와 하나 이상이 일치하면 참입니다.
ALL	메인 쿼리의 비교 조건이 서브 쿼리의 검색 결과와 모든 값이 일치하면 참입니다.
EXISTS	메인 쿼리의 비교 조건이 서브 쿼리의 결과 중에서 만족하는 값이 하나라도 존재하면 참입니다.

04. 다중 행 서브 쿼리



- ❖ 결과가 2개 이상 구해지는 쿼리문을 서브 쿼리로 기술할 경우에는 다중 행 연산자와 함께 사용해야 합니다.
- ❖ 문제 : 급여를 3000 이상 받는 사원이 소속된 부서(10번, 20번)와 동일한 부서에서 근무하는 사원구하기

```
SELECT ENAME, SAL, DEPTNO
FROM EMP
WHERE DEPTNO IN ( SELECT DISTINCT DEPTNO
                  FROM EMP
                  WHERE SAL >= 3000 );
```

- ❖ 서브 쿼리의 결과 중에서 하나라도 일치하면 참인 결과를 구하는 IN 연산자와 함께 사용되어야 합니다

04. 다중 행 서브 쿼리



- ❖ 문제 : 급여를 3000 이상 받는 사원이 소속된 부서(10번, 20번)와 동일한 부서에서 근무하는 사원구하기

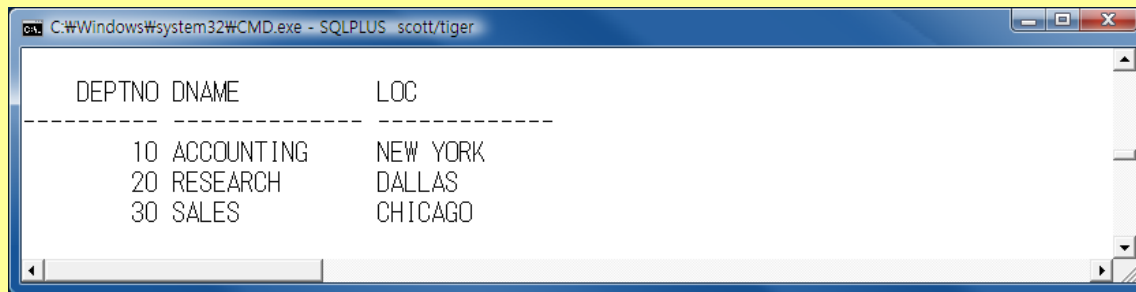
```
SELECT ENAME, SAL, DEPTNO
FROM EMP
WHERE DEPTNO IN ( SELECT DISTINCT DEPTNO
                  FROM EMP
                  WHERE SAL >= 3000 );
```

```
SELECT ENAME, SAL, DEPTNO
FROM EMP
WHERE DEPTNO IN ( 10, 20 );
```

```
SELECT ENAME, SAL, DEPTNO
FROM EMP
WHERE DEPTNO=10 OR DEPTNO=20;
```

<탄탄히 다지기>

7. 직급(JOB)이 MANAGER인 사람이 속한 부서의 부서 번호와 부서명과 지역을 출력하시오.



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\CMD.exe - SQLPLUS scott/tiger". The window displays the output of a SQL query, which is a table with three columns: DEPTNO, DNAME, and LOC. The table contains three rows of data: 10 ACCOUNTING NEW YORK, 20 RESEARCH DALLAS, and 30 SALES CHICAGO. The table is formatted with a dashed line under the column headers.

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

4.2 ALL 연산자



- ❖ 30번 소속 직원들 중에서 급여를 가장 많이 받는 직원보다 더 많은 급여를 받는 사람의 이름, 급여를 출력하는 쿼리문을 작성해 봅시다.

- ❖ 1. 단일행 서브 쿼리문

```
SELECT ENAME, SAL
FROM EMP
WHERE SAL > (SELECT MAX(SAL)
              FROM EMP
              GROUP BY DEPTNO
              HAVING DEPTNO=30);
```

ENAME	SAL
JONES	2975
SCOTT	3000
KING	5000
FORD	3000

4.2 ALL 연산자



- ❖ 30번 소속 직원들 중에서 급여를 가장 많이 받는 직원보다 더 많은 급여를 받는 사람의 이름, 급여를 출력하는 쿼리문을 작성해 봅시다.
- ❖ 2. 단일행 서브 쿼리문

```
SELECT ENAME, SAL
FROM EMP
WHERE SAL > (SELECT SAL
              FROM EMP
              WHERE DEPTNO=30)
```

단일 행 하위 질의에 2개 이상의 행이
리턴되었습니다. -에러발생
➔ 서브쿼리의 결과가 하나의 행이어야 한다.

4.2 ALL 연산자



- ❖ 30번 소속 직원들 중에서 급여를 가장 많이 받는 직원보다 더 많은 급여를 받는 사람의 이름, 급여를 출력하는 쿼리문을 작성해 봅시다.
- ❖ 3. 다중행 연산자와 함께 사용하는 경우(ALL)
 - ALL 조건은 메인 쿼리의 비교 조건이 서브 쿼리의 검색 결과와 모든 값이 일치하면 참입니다.
 - 찾아진 값에 대해서 AND 연산을 해서 모두 참이면 참이 되는 셈이 됩니다. > ALL 은 “모든 비교값 보다 크냐” 고 묻는 것이 되므로 최대값보다 더 크면 참이 됩니다.

```
SELECT ENAME, SAL
FROM EMP
WHERE SAL > ALL(SELECT SAL
                  FROM EMP
                  WHERE DEPTNO =30);
```


4.2 ALL 연산자



```
SELECT ENAME, SAL
FROM EMP
WHERE SAL > ALL(SELECT SAL
                  FROM EMP
                  WHERE DEPTNO =30);
```

❖ **select sal from emp where deptno=30;**

SAL

1600
1250
1250
2850
1500
950

```
SELECT ENAME, SAL
FROM EMP
WHERE SAL > 1600
AND SAL > 1250
AND SAL > 1250
AND SAL > 2850
AND SAL > 1500
AND SAL > 950 ;
```

<탄탄히 다지기>

8. 영업 사원들 보다 급여를 많이 받는 사원들의 이름과 급여와 직급(담당 업무)을 출력 합니다.

```
명령 프롬프트 - sqlplus scott/tiger
```

ENAME	SAL	JOB
CLARK	2450	MANAGER
BLAKE	2850	MANAGER
JONES	2975	MANAGER
SCOTT	3000	ANALYST
FORD	3000	ANALYST
KING	5000	PRESIDENT

6 rows selected.

SQL>

4.3 ANY 연산자



- ❖ ANY 조건은 메인 쿼리의 비교 조건이 서브 쿼리의 검색 결과와 하나 이상만 일치하면 참입니다.
- ❖ > ANY는 찾아진 값에 대해서 하나라도 크면 참이 되는 셈이 됩니다. 그러므로 찾아진 값 중에서 가장 작은 값 즉, 최소값 보다 크면 참이 됩니다.

4.3 ANY 연산자



- ❖ 다음은 부서번호가 30번인 직원들의 급여 중 가장 작은 값(950)보다 많은 급여를 받는 직원의 이름, 급여를 출력하는 예제를 작성해 봅시다.
- ❖ 1. 단일행 서브 쿼리문

```
SELECT ENAME, SAL
FROM EMP
WHERE SAL > ( SELECT MIN(SAL)
              FROM EMP
              GROUP BY DEPTNO
              HAVING DEPTNO = 30);
```

ENAME	SAL
---	---
--	
ALLEN	1600
WARD	1250
JONES	2975
MARTIN	1250
BLAKE	2850
CLARK	2450
SCOTT	3000
KING	5000
TURNER	1500
ADAMS	1100
FORD	3000
MILLER	1300
12 rows selected.	

4.3 ANY 연산자



- ❖ 다음은 부서번호가 30번인 직원들의 급여 중 가장 작은 값(950)보다 많은 급여를 받는 직원의 이름, 급여를 출력하는 예제를 작성해 봅시다.
 - 2. 다중행 서브 쿼리 - ANY
- ❖ ANY 조건은 메인 쿼리의 비교 조건이 서브 쿼리의 검색 결과와 하나 이상만 일치하면 참입니다.
- ❖ > ANY는 찾아진 값에 대해서 하나라도 크면 참이 되는 셈이 됩니다. 그러므로 찾아진 값 중에서 가장 작은 값 즉, **최소값 보다 크면 참**이 됩니다.

```
SELECT ENAME, SAL
FROM EMP
WHERE SAL > ANY ( SELECT SAL
                   FROM EMP
                   WHERE DEPTNO = 30 );
```

4.2 ANY 연산자



```
SELECT ENAME, SAL
FROM EMP
WHERE SAL > ANY(SELECT SAL
                  FROM EMP
                  WHERE DEPTNO =30);
```

❖ **select sal from emp where deptno=30;**

SAL

1600
1250
1250
2850
1500
950

```
SELECT ENAME, SAL
FROM EMP
WHERE SAL > 1600
OR      SAL > 1250
OR      SAL > 1250
OR      SAL > 2850
OR      SAL > 1500
OR      SAL > 950 ;
```

<탄탄히 다지기>

9. 영업 사원들의 최소 급여 보다 많이 받는 사원들의 이름과 급여와 직급(담당 업무)를 출력하되 영업 사원은 출력하지 않습니다.

```
명령 프롬프트 - sqlplus scott/tiger
```

ENAME	SAL	JOB
KING	5000	PRESIDENT
SCOTT	3000	ANALYST
FORD	3000	ANALYST
JONES	2975	MANAGER
BLAKE	2850	MANAGER
CLARK	2450	MANAGER
MILLER	1300	CLERK

7 rows selected.

SQL>

〈탄탄히 다지기〉

10. SMITH와 동일한 직급을 가진 사원의 이름과 직급을 출력하라
(단일행 서브 쿼리 이용)

ENAME	JOB
-----	-----
SMITH	CLERK
ADAMS	CLERK
JAMES	CLERK
MILLER	CLERK

11. 직급이 'SALESMAN'인 사원이 받는 급여들의 최대 급여보다 많이 받는 사원들의 이름과 급여를 출력하되 부서번호가 20번인 사원은 제외한다.

ENAME	SAL
-----	-----
CLARK	2450
BLAKE	2850
KING	5000

〈탄탄히 다지기〉

12. 직급이 'SALESMAN'인 사원이 받는 급여들의 최소 급여보다 많이 받는 사원들의 이름과 급여를 출력하되 부서번호가 20번인 사원은 제외한다.

ENAME	SAL
-----	-----
KING	5000
BLAKE	2850
CLARK	2450
ALLEN	1600
TURNER	1500
MILLER	1300

6 rows selected.

EXISTS – 특정 칼럼의 값이 존재하는지 여부를 체크

*EXISTS는 괄호 안의 쿼리(서브쿼리)에 어떤 것이든 결과 반환되면 참이 됩니다. 결과값이 반환되지 않으면 거짓으로 데이터가 추출되지 않는다.

```
SELECT ENAME, SAL
FROM EMP
WHERE EXISTS ( SELECT SAL
                FROM EMP
                WHERE DEPTNO = 30 );
```

```
SELECT ENAME, SAL
FROM EMP
WHERE EXISTS ( SELECT SAL
                FROM EMP
                WHERE DEPTNO = 40 );
```

다중 열 서브쿼리

- 서브 쿼리의 결과가 두 개 이상의 컬럼으로 반환되어 메인 쿼리에 전달되는 쿼리
- 부서별 가장 작은 급여를 받는 직원 조회

```
SELECT DEPTNO, ENAME, SAL
FROM EMP
WHERE (DEPTNO, SAL) IN (SELECT DEPTNO, MIN(SAL)
                        FROM EMP
                        GROUP BY DEPTNO)
ORDER BY DEPTNO;
```

DEPTNO	ENAME	SAL
10	MILLER	1300
20	SMITH	800
30	JAMES	950

다중 열 서브쿼리

```
SELECT DEPTNO, MIN(SAL)
FROM EMP
GROUP BY DEPTNO;
```

DEPTNO	MIN(SAL)
30	950
20	800
10	1300

```
SELECT DEPTNO, ENAME, SAL
FROM EMP
WHERE (DEPTNO=30 AND SAL=950)
OR (DEPTNO=20 AND SAL=800)
OR (DEPTNO=10 AND SAL=1300);
```

<탄탄히 다지기>

13. 부서별로 가장 급여를 많이 받는 사원의 정보(사원 번호, 사원 이름, 급여, 부서번호)를 출력하시오.(IN 연산자 이용)

EMPNO	ENAME	SAL	DEPTNO
7839	KING	5000	10
7788	SCOTT	3000	20
7902	FORD	3000	20
7698	BLAKE	2850	30

SET 연산자 - UNION

★UNION

- 두 테이블의 결합을 나타내며,
결합시키는 두 테이블의 중복 데이터는 제거한다.
- SELECT 문장의 열의 개수가 반드시 같아야 한다.
- 결합시 반드시 같은 자료형이어야 한다.
- 중복을 제거하기 위해 SORT 함
- 자료가 많거나 INDEX가 되어있지 않는 컬럼을 대상으로 하면
쿼리시간이 길어질수 있음

★UNION ALL : 두 테이블의 중복되는 값까지 반환한다.
SORT 안됨

SET 연산자 - UNION(합집합)

1. SET 연산자를 사용할 경우에는 SELECT 문장의 열의 개수가 반드시 같아야 한다.

```
SQL> SELECT 1, 3, 5  
2 FROM DUAL  
3 UNION  
4 SELECT 2, 4, 6  
5 FROM DUAL;
```

1	3	5
1	3	5
2	4	6

SET 연산자 - UNION(합집합)

2. 결합시키는 두 테이블의 중복 데이터는 제거한다.

```
SQL> SELECT ENAME  
2     FROM EMP  
3     UNION  
4     SELECT ENAME  
5     FROM EMP;
```

3. 결합시 반드시 같은 자료형이어야 한다.

```
SQL> SELECT ENAME  
2     FROM EMP  
3     UNION  
4     SELECT DEPTNO  
5     FROM EMP;
```

ERROR at line 1:

ORA-01790: expression must have same datatype as corresponding expression

SET 연산자 - UNION ALL

결합시키는 두 테이블의 중복 데이터까지 반환한다.

```
SQL> SELECT ENAME  
2     FROM EMP  
3     UNION ALL  
4     SELECT ENAME  
5     FROM EMP;
```

SET 연산자 - INTERSECT

여러 개의 SQL문의 결과에 대한 교집합을 나타내고 중복 행은 하나의 결과로 보여줍니다.

```
SQL> SELECT empno, ename, sal  
      FROM emp  
      WHERE sal > 1000  
      INTERSECT  
      SELECT empno, ename, sal  
      FROM emp  
      WHERE sal < 2000
```

SET 연산자 - MINUS

여러 개의 SQL문의 결과에 대한 차집합을 나타냅니다.
공통된 항목을 제외한 결과만 추출합니다.

```
SQL> SELECT empno, ename, sal  
      FROM emp  
      WHERE sal > 1000  
      MINUS  
      SELECT empno, ename, sal  
      FROM emp  
      WHERE sal < 2000
```