



SQL

✓ 원리를 알면 IT가 맛있다

SQL for Beginners

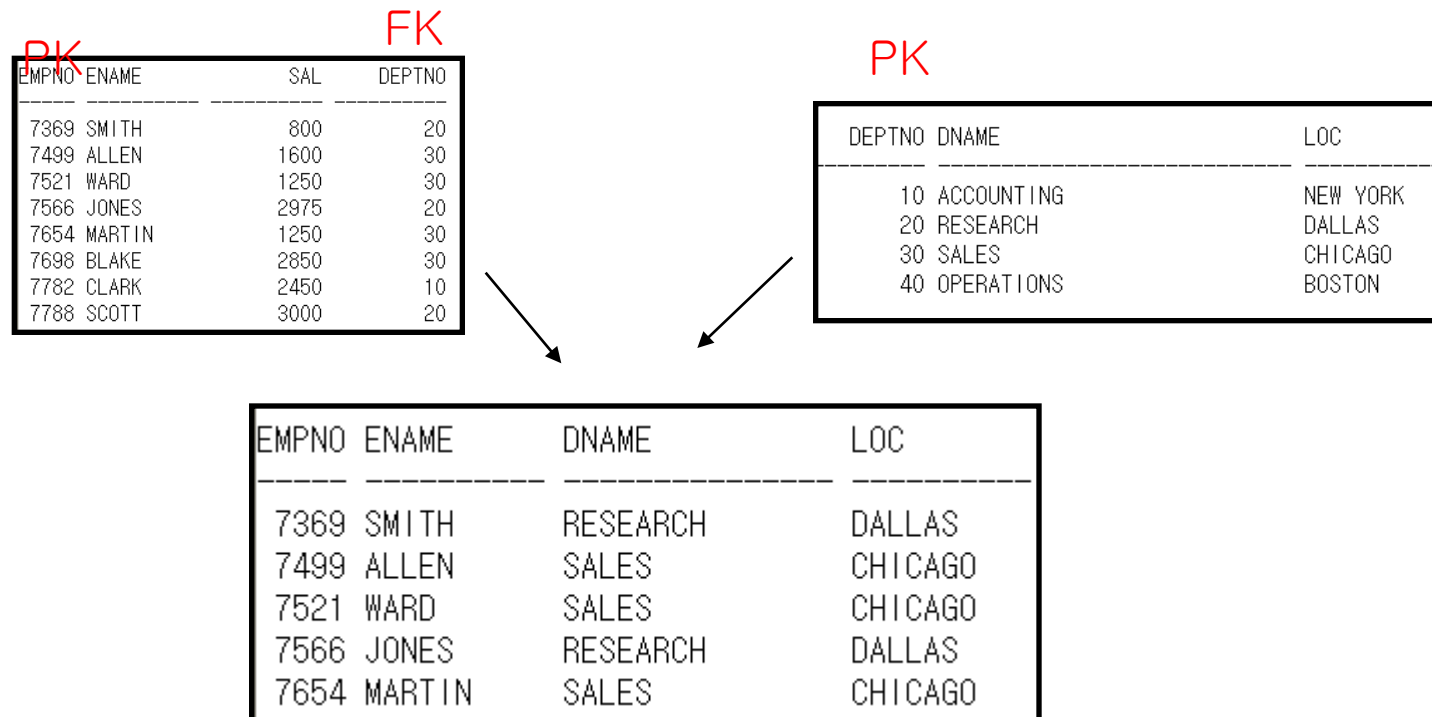
chapter 06.

조인

- 조인 개념
- 오라클 조인 종류
 - catasian product
 - equi 조인
 - non-equi 조인
 - outer 조인
 - self 조인
- ANSI 조인 종류
 - cross Join
 - natural Join
 - using(컬럼) 및 join~on 절
 - left|right|full outer join

■ JOIN 정의

- 검색하고자 하는 컬럼이 한 개의 테이블이 아닌, 여러 개의 테이블에 존재하는 경우에 사용되는 기술.
- 일반적으로 Primary Key(이후 PK 로 사용)와 Foreign Key(이후 FK 로 사용)를 사용하여 Join 하는 경우가 대부분이지만 때로는 논리적인 값들의 연관으로 Join 하는 경우도 있다.



- JOIN 종류

- 1. 오라클 조인 (Oracle 8i)

- 카테시안 프로덕트 (Catasian Product)
 - Equi-Join
 - Non-EquiJoin
 - Outer 조인
 - Self 조인

- 2. ANSI 조인 (Oracle 9i 이후)

- Cross 조인
 - Natural 조인
 - Using 이용한 조인
 - Outer 조인

■ Catasian Product

모든 가능한 행들의 Join 으로 다음과 같은 경우에 발생한다.

- 1) 조인 조건이 생략된 경우
- 2) 조인 조건이 잘못된 경우
- 3) 첫번째 테이블의 모든 행이 두번째 테이블의 모든 행과 두번째 테이블의 모든 행이 첫번째 테이블의 모든 행과 조인되는 경우.
- 4) 양쪽 ROW 의 개수를 곱한 결과

```
SQL> SELECT empno, ename, job, dept.deptno, dname, loc  
2 FROM dept, emp  
3 ORDER BY empno;
```

■ Equi 조인

Equi 조인이란 조인 조건에서 "="를 사용하여 값들이 정확하게 일치하는 경우에 사용하는 조인을 말한다. 대부분 PK와 FK의 관계를 이용하여 조인 하며, Equi 조인은 다른 말로 단순 조인 또는 내부 조인(inner) 이라고도 한다.

```
SELECT table1.column, table2.column  
FROM table1, table2  
WHERE table1.column1 = table2.column2;
```

■ Equi 조인 방법

- 1) SELECT절은 검색할 열 이름을 명시
- 2) FROM절은 데이터베이스가 Access해야 하는 두 개의 테이블을 명시
- 3) WHERE절은 테이블의 조인 조건을 명시
- 4) 양쪽 테이블에 공통으로 존재하는 열 이름은 모호함을 피하기 위하여 열 이름 앞에 테이블명을 기술함.

```
SQL> SELECT EMP.EMPNO , EMP.ENAME , DEPT.DNAME, DEPT.LOC
2 FROM EMP , DEPT
3 WHERE EMP.DEPTNO = DEPT.DEPTNO;
```

EMPNO	ENAME	DNAME	LOC
7369	SMITH	RESEARCH	DALLAS
7499	ALLEN	SALES	CHICAGO
7521	WARD	SALES	CHICAGO

```
SQL> SELECT e.EMPNO , e.ENAME , d.DNAME, d.LOC
2 FROM EMP e, DEPT d
3 WHERE e.DEPTNO = d.DEPTNO;
```

EMPNO	ENAME	DNAME	LOC
7369	SMITH	RESEARCH	DALLAS
7499	ALLEN	SALES	CHICAGO
7521	WARD	SALES	CHICAGO



```
SQL> SELECT e.EMPNO , e.ENAME , d.DNAME, d.LOC
2 FROM EMP e, DEPT d
3 WHERE e.DEPTNO = d.DEPTNO
4 AND e.EMPNO = 7900;
```

EMPNO	ENAME	DNAME	LOC
7900	JAMES	SALES	CHICAGO

* 만약 alias를 적용해서 SQL문장을 작성한 경우에는
WHERE절과 SELECT절에는 테이블명이 아닌 alias를 사용해야 된다.

■ 두 개 이상의 테이블 조인

때로는 두 개 이상의 테이블을 조인 할 경우가 있다.

문제) 고객의 TKB SPORT SHOP의 이름, 주문처, 항목수, 각 항목의 합계, 각 주문의 합계를 출력 ?

<pre>SQL> SELECT name, custid 2 FROM customer;</pre>	<pre>SQL> SELECT ordid, itemid 2 FROM item;</pre>	<pre>SQL> SELECT custid, ordid 2 FROM ord;</pre>
<pre>NAME CUSTID ----- JOCKSPORTS 100 TKB SPORT SHOP 101 VOLLYRITE 102 9 rows selected.</pre>	<pre>ORDID ITEMID ----- 610 3 611 1 612 1 64 rows selected.</pre>	<pre>CUSTID ORDID ----- 101 610 102 611 104 612 21 rows selected.</pre>

```
SQL> SELECT c.name, o.ordid, i.itemid, i.itemtot, o.total
  2  FROM customer c, ord o, item i
  3  WHERE c.custid = o.custid AND o.ordid = i.ordid
  4  AND c.name = 'TKB SPORT SHOP';
```

NAME	ORDID	ITEMID	ITEMTOT	TOTAL
-----	-----	-----	-----	-----
TKB SPORT SHOP	610	3	58	101.4
TKB SPORT SHOP	610	1	35	101.4
TKB SPORT SHOP	610	2	8.4	101.4

■ Non-Equi 조인

EMP와 SALGRADE 사이의 관련성은 EMP 테이블의 어떠한 column도 직접적으로 SALGRADE 테이블의 한 column에 상응하지 않기 때문에 Non-Equijoin이다. 두 테이블 사이의 관련성은 EMP 테이블의 SAL열이 SALGRADE 테이블의 LOSAL과 HISAL열 사이에 있다는 것이다. 조인 조건은 등등(=) 이외의 연산자(BETWEEN ~ AND ~ , > , >= , < , <=)를 갖는다.

EMPNO	ENAME	SAL
7369	SMITH	800
7499	ALLEN	1600
7521	WARD	1250
7566	JONES	2975
7654	MARTIN	1250
7698	BLAKE	2850
7782	CLARK	2450
7788	SCOTT	3000
7839	KING	5000
7844	TURNER	1500
7876	ADAMS	1100

GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

문제) EMP 테이블에서 사원번호, 이름, 업무, 급여, 급여의 등급, 하한값, 상한값을 출력 ?

```
SQL> SELECT e.empno, e.ename, e.job, e.sal, s.grade, s.losal, s.hisal  
2 FROM salgrade s, emp e  
3 WHERE e.sal BETWEEN s.losal AND s.hisal AND e.deptno = 10;
```

EMPNO	ENAME	JOB	SAL	GRADE	LOSAL	HISAL
7839	KING	PRESIDENT	5000	5	3001	9999
7782	CLARK	MANAGER	2450	4	2001	3000
7934	MILLER	CLERK	1300	2	1201	1400

■ Outer 조인

행이 조인 조건을 만족하지 않으면, 행은 질의 결과에 나타나지 않는다.
예를 들어 EMP와DEPT테이블의 equi조인 조건에서 부서OPERATIONS(40번 부서)는 해당 부서에 아무도 없기 때문에 나타나지 않는다. 이런 경우 모든 행을 전부 출력하고자 할 경우 Outer Join을 사용한다. 즉 ,정상적으로 조인 조건을 만족하지 못하는 행들을 보기 위해 Outer join을 사용한다.

EMPNO	ENAME	DEPTNO
7369	SMITH	20
7499	ALLEN	30
7521	WARD	30
7566	JONES	20
7654	MARTIN	30
7698	BLAKE	30
7782	CLARK	10
7788	SCOTT	20
7839	KING	10
7844	TURNER	30
7876	ADAMS	20
<hr/>		
EMPNO	ENAME	DEPTNO
7900	JAMES	30
7902	FORD	20
7934	MILLER	10

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

↓
40 은 항상 누락된다.

■ Outer 조인

- 행인 조건을 만족하지 않을 시 해당 행은 질의 결과에 나타나지 않는다.
- Outer join 연산자를 조인 조건에 사용시 조인 조건을 만족하지 않는 행들도 결과에 나타날 수 있다.
- 연산자는 괄호로 묶인 플러스 기호(+)이며 조인 시킬 값이 없는 조인 측에 (+)를 위치 시킨다.
- (+)연산자는 한 개 이상의 NULL 행을 생성하고 정보가 충분한 테이블의 한 개 이상의 행들이 이런 NULL 행에 조인된다.
- Outer join 연산자는 표현식의 한 편에만 올 수 있다.
- Outer join을 포함하는 조건은 IN 연산자, OR 연산자를 사용하여 다른 하나의 조건에 연결될 수 없다. 단, IN 연산자의 값이 한 개인 경우에는 사용가능.

문제) EMP 테이블과 DEPT 테이블에서 DEPT 테이블에 있는 모든 자료를 사
원번호, 이름, 업무, EMP 테이블의 부서번호, DEPT 테이블의 부서번호, 부서
명, 근무지를 출력 ?

```
SQL> SELECT e.empno, e.ename, e.job, e.deptno,
2  d.deptno, d.dname, d.loc
3  FROM dept d,emp e
4  WHERE d.deptno = e.deptno(+);
```

EMPNO	ENAME	JOB	DEPTNO	DEPTNO	DNAME	LOC
7900	JAMES	CLERK	30	30	SALES	CHICAGO
7521	WARD	SALESMAN	30	30	SALES	CHICAGO
				40	OPERATIONS	BOSTON

15 rows selected.

■ Self 조인

- 특정 테이블 자신을 자신이 Join하는 방법이다.
- 때때로 자체적으로 테이블을 조인할 필요가 있다. 각 종업원의 관리자 명을 알기 위해서 자체적으로 EMP테이블을 조인하는 것이 필요하다.

■ Self 조인 가이드라인

- Self join을 사용하여 한 테이블의 행들을 같은 테이블의 행들과 조인한다.
- 같은 테이블에 대해 두 개의 alias를 작성(테이블 구분)함으로 FROM절에 두 개의 테이블을 사용하는 것과 같이 한다.
- Column에 대해서도 어떤 테이블에서 왔는지 반드시 Alias명을 기술하여야 한다.
- 테이블 하나를 두 개 또는 그 이상으로 Self join할 수 있다.


```
SQL> SELECT EMPNO, ENAME , MGR
2 FROM EMP;
```

EMPNO	ENAME	MGR
7369	SMITH	7902
7499	ALLEN	7698
7521	WARD	7698
7566	JONES	7839
7654	MARTIN	7698
7698	BLAKE	7839
7782	CLARK	7839
7788	SCOTT	7566
7839	KING	
7844	TURNER	7698
7876	ADAMS	7788
<hr/>		
EMPNO	ENAME	MGR
7900	JAMES	7698
7902	FORD	7566
7934	MILLER	7782

```
SQL> SELECT EMPNO, ENAME
2 FROM EMP;
```

EMPNO	ENAME
7369	SMITH
7499	ALLEN
7521	WARD
7566	JONES
7654	MARTIN
7698	BLAKE
7782	CLARK
7788	SCOTT
7839	KING
7844	TURNER
7876	ADAMS
<hr/>	
EMPNO	ENAME
7900	JAMES
7902	FORD
7934	MILLER

SMITH의 관리자는 FORD

문제) EMP 테이블에서 Self join하여 관리자를 출력 ?

```
SQL> SELECT a.ENAME 사원 , b.ENAME 관리자
2 FROM EMP a , EMP b
3 WHERE a.MGR = b.EMPNO;
```

사원	관리자
SMITH	FORD
ALLEN	BLAKE
WARD	BLAKE
JONES	KING
MARTIN	BLAKE
BLAKE	KING
CLARK	KING
SCOTT	JONES
TURNER	BLAKE
ADAMS	SCOTT
JAMES	BLAKE

사원	관리자
FORD	JONES
MILLER	CLARK

13 개의 행이 선택되었습니다.

```
SQL> SELECT a.ENAME 사원 , b.ENAME 관리자
2 FROM EMP a , EMP b
3 WHERE a.MGR = b.EMPNO(+);
```

사원	관리자
SMITH	FORD
ALLEN	BLAKE
WARD	BLAKE
JONES	KING
MARTIN	BLAKE
BLAKE	KING
CLARK	KING
SCOTT	JONES
KING	
TURNER	BLAKE
ADAMS	SCOTT

사원	관리자
JAMES	BLAKE
FORD	JONES
MILLER	CLARK

14 개의 행이 선택되었습니다.

- ANSI 조인 특징
 - Join의 형식이 FROM 절에서 지정된다.
 - Join 조건이 WHERE 절이 아닌 ON절에서 명시된다.
(이전에는 WHERE절 안에 JOIN조건과 검색조건이 같이 사용되어 용도가 불분명한 경우가 발생됨.)
 - NATURAL JOIN과 USING절에서는 공통 컬럼명에 alias 사용 못함.

```
SELECT table1.column, table2.column
FROM table1
[CROSS JOIN table2] |
[NATURAL JOIN table2] |
[JOIN table2 USING (column_name)] |
[JOIN table2
  ON (table1.column_name = table2.column_name)] |
[LEFT|RIGHT|FULL OUTER JOIN table2
  ON (table1.column_name = table2.column_name)];
```

- Cross Join
: Catasian Product와 동일하다.

```
SQL> SELECT EMPNO , ENAME , DNAME  
2 FROM DEPT  
3 CROSS JOIN EMP;
```

EMPNO	ENAME	DNAME
7369	SMITH	ACCOUNTING
7499	ALLEN	ACCOUNTING
7521	WARD	ACCOUNTING

```
SQL> SELECT EMPNO, ENAME, DNAME  
2 FROM DEPT, EMP;
```

EMPNO	ENAME	DNAME
7369	SMITH	ACCOUNTING
7499	ALLEN	ACCOUNTING
7521	WARD	ACCOUNTING

- Natural Join
 - : Equi-Join 과 동일하다. 즉, 같은 이름을 가진 컬럼에 기반한다.
 - : Natural Join을 사용할 때의 전제조건은 양쪽 Table에 반드시 1개만의 공통 컬럼명만이 있어야 한다. 2 개 이상의 공통 컬럼이 있다면 오류는 아니지만 엉뚱한 결과를 가져온다.
 - 즉, 2개의 공통 컬럼 값이 서로 같은 것만 추출해 온다.

SQL> SELECT EMPNO , ENAME, DNAME, LOC			
2 FROM EMP			
3 <u>NATURAL JOIN DEPT;</u>			
EMPNO	ENAME	DNAME	LOC
<hr/>			
7369	SMITH	RESEARCH	DALLAS
7499	ALLEN	SALES	CHICAGO
7521	WARD	SALES	CHICAGO

SQL> SELECT e.EMPNO , e.ENAME , d.DNAME, d.LOC			
2 FROM EMP e, DEPT d			
3 WHERE e.DEPTNO = d.DEPTNO;			
EMPNO	ENAME	DNAME	LOC
<hr/>			
7369	SMITH	RESEARCH	DALLAS
7499	ALLEN	SALES	CHICAGO
7521	WARD	SALES	CHICAGO

■ USING(컬럼)

- 동일이름의 컬럼이 여러 개인 경우 조인 컬럼을 지정.
- Natural Join과는 상호배타적이다.
- 여기서 컬럼명은 조회할 자료가 있는 테이블에 공통된 컬럼명이다.
NATURAL JOIN과 USING 절에서는 공통 컬럼명에 Table Alias 명을 사용하지 못한다.

```
SQL> SELECT EMPNO, ENAME, DNAME, LOC  
2 FROM EMP  
3 JOIN DEPT  
4 USING (DEPTNO);
```

EMPNO	ENAME	DNAME	LOC
7369	SMITH	RESEARCH	DALLAS
7499	ALLEN	SALES	CHICAGO
7521	WARD	SALES	CHICAGO

■ join~on 이용한 Join

- Non-Equi조인 또는 임의의 조건으로 조인시 사용.
- 조인할 컬럼을 명시하기 위해서 사용.
- 복잡한 조건의 조인 가능하다.

(서브쿼리 , AND/OR 연산자 , EXIST , IN 연산자)

- ON 조건절에 JOIN 조건 외에도 데이터 검색조건을 추가할 수는 있으나, 검색 조건 목적인 경우는 WHERE절을 사용함을 권장한다.

```
SQL> SELECT EMPNO, ENAME, DNAME, LOC
2 FROM EMP
3 JOIN DEPT
4 ON EMP.DEPTNO = DEPT.DEPTNO
5 AND DEPT.DEPTNO = 10;
```

EMPNO	ENAME	DNAME	LOC
7782	CLARK	ACCOUNTING	NEW YORK
7839	KING	ACCOUNTING	NEW YORK
7934	MILLER	ACCOUNTING	NEW YORK



```
SQL> SELECT EMPNO, ENAME, DNAME, LOC
2 FROM EMP
3 JOIN DEPT
4 ON EMP.DEPTNO = DEPT.DEPTNO
5 WHERE DEPT.DEPTNO = 10;
```

EMPNO	ENAME	DNAME	LOC
7782	CLARK	ACCOUNTING	NEW YORK
7839	KING	ACCOUNTING	NEW YORK
7934	MILLER	ACCOUNTING	NEW YORK

```
SQL> SELECT EMPNO , ENAME, DNAME, SAL , GRADE
2 FROM EMP e
3 JOIN DEPT d
4 ON e.DEPTNO = d.DEPTNO
5 JOIN SALGRADE s
6 ON e.SAL BETWEEN s.LOSAL AND s.HISAL;
```

EMPNO	ENAME	DNAME	SAL	GRADE
7369	SMITH	RESEARCH	800	1
7900	JAMES	SALES	950	1
7876	ADAMS	RESEARCH	1100	1
7521	WARD	SALES	1250	2
7654	MARTIN	SALES	1250	2
7934	MILLER	ACCOUNTING	1300	2
7844	TURNER	SALES	1500	3

```
SQL> SELECT EMPNO , ENAME, DNAME, SAL
2 FROM EMP e
3 JOIN DEPT d
4 ON e.DEPTNO = d.DEPTNO
5 AND SAL IN ( 800 );
```

EMPNO	ENAME	DNAME	SAL
7369	SMITH	RESEARCH	800

```
SQL> SELECT e.ename 사원 , d.ename 관리자
2 FROM emp e
3 join emp d
4 on e.mgr = d.empno;
```

사원	관리자
SMITH	FORD

- LEFT OUTER JOIN ~ ON | USING
 - 좌측에 기술한 테이블의 모든 행들이 우측에 기술한 테이블내 행들과 일치 여부에 상관없이 모두 출력된다.
- RIGHT OUTER JOIN ~ ON | USING
 - 우측에 기술한 테이블의 모든 행들이 좌측에 기술한 테이블내 행들과 일치 여부에 상관없이 모두 출력된다.
- FULL OUTER JOIN ~ ON | USING
 - LEFT JOIN과 RIGHT JOIN 결과를 합집합으로 처리한 결과와 동일.
(UNION)

□ 4 ANSI 조인 – Outer Join

```
SQL> SELECT EMPNO ,ENAME , DNAME
2 FROM DEPT LEFT OUTER JOIN EMP
3 ON DEPT.DEPTNO = EMP.DEPTNO;
```

EMPNO	ENAME	DNAME
7369	SMITH	RESEARCH
7499	ALLEN	SALES
7521	WARD	SALES
7566	JONES	RESEARCH
7654	MARTIN	SALES
7698	BLAKE	SALES
7782	CLARK	ACCOUNTING
7788	SCOTT	RESEARCH
7839	KING	ACCOUNTING
7844	TURNER	SALES
7876	ADAMS	RESEARCH

EMPNO	ENAME	DNAME
7900	JAMES	SALES
7902	FORD	RESEARCH
7934	MILLER	ACCOUNTING
		OPERATIONS

15 개의 행이 선택되었습니다.

```
SQL> INSERT INTO EMP
2 VALUES ( 9000, 'TEST' , 'SALES',7499,'90/01/01', 400 , NULL , NULL );
```

1 개의 행이 만들어졌습니다.

```
SQL> SELECT EMPNO, ENAME, DNAME
2 FROM DEPT FULL OUTER JOIN EMP
3 ON DEPT.DEPTNO = EMP.DEPTNO;
```

EMPNO	ENAME	DNAME
7369	SMITH	RESEARCH
7499	ALLEN	SALES
7521	WARD	SALES
7566	JONES	RESEARCH
7654	MARTIN	SALES
7698	BLAKE	SALES
7782	CLARK	ACCOUNTING
7788	SCOTT	RESEARCH
7839	KING	ACCOUNTING
7844	TURNER	SALES
7876	ADAMS	RESEARCH

EMPNO	ENAME	DNAME
7900	JAMES	SALES
7902	FORD	RESEARCH
7934	MILLER	ACCOUNTING
		OPERATIONS
9000	TEST	

16 개의 행이 선택되었습니다.

■ 실습 문제 (Oracle 조인 및 ANSI 조인 모두 구현)

1. 부서 테이블과 사원테이블에서 사번, 사원명, 부서코드, 부서명을 검색하시오.
(사원명 오름차순 정렬할 것)
2. 부서 테이블과 사원테이블에서 사번, 사원명 , 급여 , 부서명을 검색하시오. 단, 급여가 2000 이상인 사원에 대하여 급여기준으로 내림차순 정렬할 것.
3. 부서 테이블과 사원 테이블에서 사번, 사원명, 업무, 급여 , 부서명을 검색하시오. 단, 업무가 Manager이며 급여가 2500 이상인 사원에 대하여 사번을 기준으로 오름차순 정렬할 것.
4. 사원 테이블과 급여 등급 테이블에서 사번, 사원명, 급여, 등급을 검색하시오. 단, 등급은 급여가 하한값과 상한값 범위에 포함되고 등급이 4이며 급여를 기준으로 내림차순정렬할 것.
5. 부서 테이블, 사원 테이블, 급여등급 테이블에서 사번, 사원명, 부서명, 급여 , 등급을 검색하시오. 단, 등급은 급여가 하한값과 상한값 범위에 포함되며 등급을 기준으로 내림차순 정렬할 것.
6. 사원 테이블에서 사원명과 해당 사원의 관리자명을 검색하시오.
7. 사원 테이블에서 사원명, 해당 사원의 관리자명, 해당 사원의 관리자의 관리자명을 검색하시오
8. 7번 결과에서 상위 관리자가 없는 모든 사원의 이름도 사원명에 출력되도록 수정하시오.
9. 20번 부서의 이름과 그 부서에 근무하는 사원의 이름을 출력하시오.

10. 커미션을 받는 사원의 이름, 커미션, 부서이름 출력하시오.
11. 이름에 'A' 가 들어가는 사원들의 이름과 부서명 출력하기.
12. DALLAS에 근무하는 사원 중 급여 1500 이상인 사원의 이름, 급여, 입사일 , 보너스(comm)을 출력하시오.
13. 자신의 관리자 보다 연봉(sal)을 많이 받는 사원의 이름과 연봉을 출력하시오.
14. 직원 중 현재시간 기준으로 근무 개월 수가 30년($12 * 30$ 개월) 보다 많은 사람의 이름, 급여 , 입사일 , 부서명을 출력하시오.
15. 각 부서별로 1982년 이전에 입사한 직원들의 인원수를 출력하시오.



Thank you
