

## <DataBase>





1. 다음의 모든 조건을 만족하는 SQL 문을 작성하세요.

1) 조회할 모든 테이블은 EMP 테이블이며 모든 열을 출력합니다.

SELECT \* FROM EMP;

스크립트 출력 x

질의 결과 x

    SQL | 인출된 모든 행: 12(0,002초)

|    | EMPNO | ENAME  | JOB       | MGR    | HIREDATE   | SAL  | COMM   | DEPTNO |
|----|-------|--------|-----------|--------|------------|------|--------|--------|
| 1  | 7369  | SMITH  | CLERK     | 7902   | 1980/12/17 | 800  | (null) | 20     |
| 2  | 7499  | ALLEN  | SALESMAN  | 7698   | 1981/02/20 | 1600 | 300    | 30     |
| 3  | 7521  | WARD   | SALESMAN  | 7698   | 1981/02/22 | 1250 | 500    | 30     |
| 4  | 7566  | JONES  | MANAGER   | 7839   | 1981/04/02 | 2975 | (null) | 20     |
| 5  | 7654  | MARTIN | SALESMAN  | 7698   | 1981/09/28 | 1250 | 1400   | 30     |
| 6  | 7698  | BLAKE  | MANAGER   | 7839   | 1981/05/01 | 2850 | (null) | 30     |
| 7  | 7782  | CLARK  | MANAGER   | 7839   | 1981/06/09 | 2450 | (null) | 10     |
| 8  | 7839  | KING   | PRESIDENT | (null) | 1981/11/17 | 5000 | (null) | 10     |
| 9  | 7844  | TURNER | SALESMAN  | 7698   | 1981/09/08 | 1500 | 0      | 30     |
| 10 | 7900  | JAMES  | CLERK     | 7698   | 1981/12/03 | 950  | (null) | 30     |
| 11 | 7902  | FORD   | ANALYST   | 7566   | 1981/12/03 | 3000 | (null) | 20     |
| 12 | 7934  | MILLER | CLERK     | 7782   | 1982/01/23 | 1300 | (null) | 10     |

2) 출력되는 열의 별칭은 다음과 같습니다.

EMPNO열 ► EMPLOYEE\_NO

ENAME열 ► EMPLOYEE\_NAME

MGR열 ► MANAGER

SAL열 ► SALARY

COMM열 ► COMMISSION

DEPTNO열 ► DEPARTMENT\_NO

3) 부서 번호를 기준으로 내림차순 정렬하되 부서 번호가 같다면 사원 이름을 기준으로 오름차순 정렬합니다.

```
SELECT EMPNO AS EMPLOYEE_NO, ENAME AS EMPLOYEE_NAME, MGR AS MANGER, SAL
AS SALARY, COMM AS COMMISSION,
DEPTNO AS DEPARTMENT_NO FROM EMP ORDER BY DEPARTMENT_NO DESC, ENAME;
```

스크립트 출력 x

질의 결과 x

SQL

인출된 모든 행: 12(0.003초)

|    | EMPLOYEE_NO | EMPLOYEE_NAME | MANGER | SALARY | COMMISSION | DEPARTMENT_NO |
|----|-------------|---------------|--------|--------|------------|---------------|
| 1  | 7499        | ALLEN         | 7698   | 1600   | 300        | 30            |
| 2  | 7698        | BLAKE         | 7839   | 2850   | (null)     | 30            |
| 3  | 7900        | JAMES         | 7698   | 950    | (null)     | 30            |
| 4  | 7654        | MARTIN        | 7698   | 1250   | 1400       | 30            |
| 5  | 7844        | TURNER        | 7698   | 1500   | 0          | 30            |
| 6  | 7521        | WARD          | 7698   | 1250   | 500        | 30            |
| 7  | 7902        | FORD          | 7566   | 3000   | (null)     | 20            |
| 8  | 7566        | JONES         | 7839   | 2975   | (null)     | 20            |
| 9  | 7369        | SMITH         | 7902   | 800    | (null)     | 20            |
| 10 | 7782        | CLARK         | 7839   | 2450   | (null)     | 10            |
| 11 | 7839        | KING          | (null) | 5000   | (null)     | 10            |
| 12 | 7934        | MILLER        | 7782   | 1300   | (null)     | 10            |

2. EMP 테이블을 사용하여 30번 부서(DEPTNO)에서 근무하고 있는 사원 중에 직책(JOB)이 SALESMAN인 사원의 사원번호, 이름, 직책, 급여, 부서번호를 출력하는 SQL문을 작성하세요.

```
SELECT EMPNO, ENAME, JOB, SAL, DEPTNO FROM EMP
WHERE DEPTNO = 30
AND JOB = 'SALESMAN';
```

스크립트 출력 x

질의 결과 x

SQL | 인출된 모든 행: 4(0.003초)

|   | EMPNO | ENAME  | JOB      | SAL  | DEPTNO |
|---|-------|--------|----------|------|--------|
| 1 | 7499  | ALLEN  | SALESMAN | 1600 | 30     |
| 2 | 7521  | WARD   | SALESMAN | 1250 | 30     |
| 3 | 7654  | MARTIN | SALESMAN | 1250 | 30     |
| 4 | 7844  | TURNER | SALESMAN | 1500 | 30     |

3. 추가 수당이 존재하지 않고 상급자가 있고 직책이 MANAGER, CLERK인 사원 중에서 사원 이름의 두 번째 글자가 L이 아닌 사원의 정보를 출력하는 SQL문을 작성해 보세요.

```
SELECT * FROM EMP
WHERE COMM IS NULL
AND MGR IS NOT NULL
AND JOB IN ('MANAGER', 'CLERK')
AND ENAME NOT LIKE '_L%';
```

| EMPNO | ENAME       | JOB     | MGR  | HIREDATE   | SAL  | COMM   | DEPTNO |
|-------|-------------|---------|------|------------|------|--------|--------|
| 1     | 7369 SMITH  | CLERK   | 7902 | 1980/12/17 | 800  | (null) | 20     |
| 2     | 7566 JONES  | MANAGER | 7839 | 1981/04/02 | 2975 | (null) | 20     |
| 3     | 7900 JAMES  | CLERK   | 7698 | 1981/12/03 | 950  | (null) | 30     |
| 4     | 7934 MILLER | CLERK   | 7782 | 1982/01/23 | 1300 | (null) | 10     |

4. 다음과 같은 결과가 나오도록 SQL 문을 작성해보세요.

EMP 테이블에서 직원들의 월 평균 근무일 수는 21.5일입니다. 하루 근무 시간을 8시간으로 보았을 때 직원들의 하루 급여(DAY\_PAY)와 시급(TIME\_PAY)을 계산하여 결과를 출력합니다. 단 하루 급여는 소수점 세 번째 자리에서 버리고, 시급은 두 번째 소수점에서 반올림하세요.

```
SELECT EMPNO, ENAME, SAL,
TRUNC(SAL / 21.5, 2) AS DAY_PAY,
ROUND(SAL / 21.5 / 8, 1) AS TIME_PAY
FROM EMP;
```

| EMPNO | ENAME       | SAL  | DAY_PAY | TIME_PAY |
|-------|-------------|------|---------|----------|
| 1     | 7369 SMITH  | 800  | 37.2    | 4.7      |
| 2     | 7499 ALLEN  | 1600 | 74.41   | 9.3      |
| 3     | 7521 WARD   | 1250 | 58.13   | 7.3      |
| 4     | 7566 JONES  | 2975 | 138.37  | 17.3     |
| 5     | 7654 MARTIN | 1250 | 58.13   | 7.3      |
| 6     | 7698 BLAKE  | 2850 | 132.55  | 16.6     |
| 7     | 7782 CLARK  | 2450 | 113.95  | 14.2     |
| 8     | 7839 KING   | 5000 | 232.55  | 29.1     |
| 9     | 7844 TURNER | 1500 | 69.76   | 8.7      |
| 10    | 7900 JAMES  | 950  | 44.18   | 5.5      |
| 11    | 7902 FORD   | 3000 | 139.53  | 17.4     |
| 12    | 7934 MILLER | 1300 | 60.46   | 7.6      |

5. 오른쪽과 같은 결과가 나오도록 SQL문을 작성해보세요.

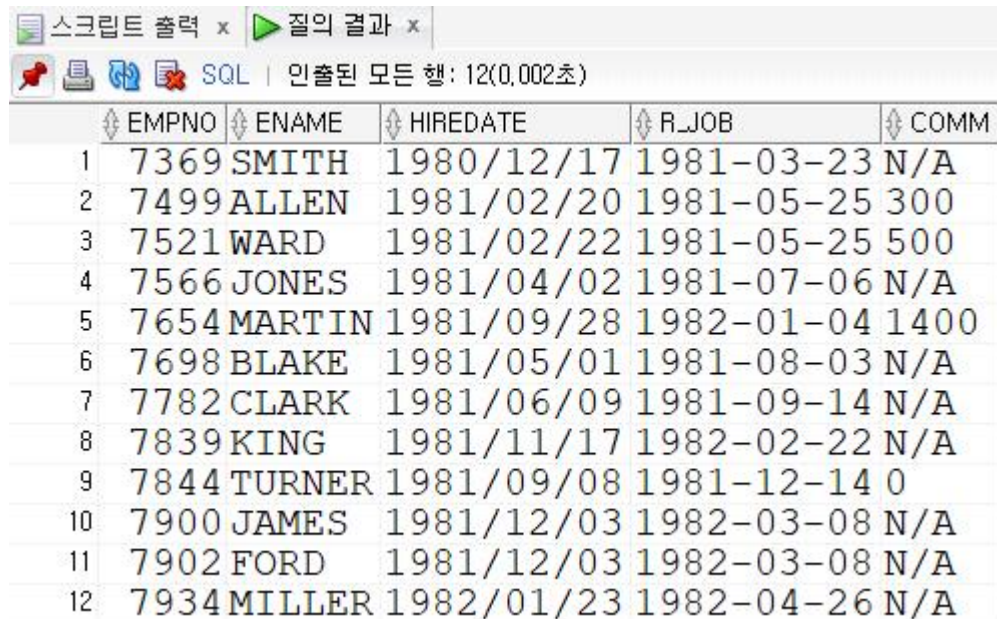
EMP 테이블에서 직원들은 입사일(HIREDATE)을 기준으로 3개월이 지난 후 첫 월요일에 정직원이 됩니다.

직원들이 정직원이 되는 날짜(R\_JOB)을 YYYY-MM-DD 형식으로 오른쪽과 같이 출력해 주세요. 단, 추가 수당(COMM)이 없는 사원의 추가수당은 N/A로 출력하세요.

```
SELECT EMPNO, ENAME, HIREDATE,
TO_CHAR(NEXT_DAY(ADD_MONTHS(HIREDATE, 3), '월요일'), 'YYYY-MM-DD') AS R_JOB,
NVL(TO_CHAR(COMM), 'N/A')AS COMM
```



FROM EMP;



스크립트 출력 x | 질의 결과 x

SQL | 인출된 모든 행: 12(0.002초)

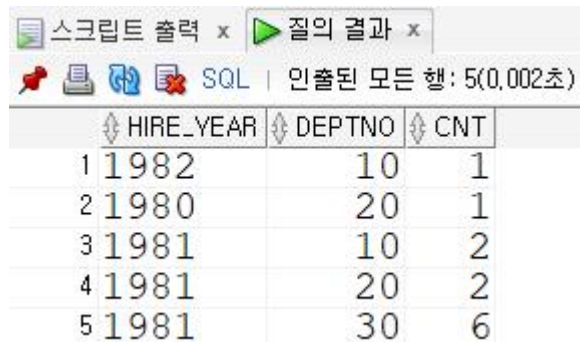
|    | EMPNO | ENAME  | HIREDATE   | R_JOB      | COMM |
|----|-------|--------|------------|------------|------|
| 1  | 7369  | SMITH  | 1980/12/17 | 1981-03-23 | N/A  |
| 2  | 7499  | ALLEN  | 1981/02/20 | 1981-05-25 | 300  |
| 3  | 7521  | WARD   | 1981/02/22 | 1981-05-25 | 500  |
| 4  | 7566  | JONES  | 1981/04/02 | 1981-07-06 | N/A  |
| 5  | 7654  | MARTIN | 1981/09/28 | 1982-01-04 | 1400 |
| 6  | 7698  | BLAKE  | 1981/05/01 | 1981-08-03 | N/A  |
| 7  | 7782  | CLARK  | 1981/06/09 | 1981-09-14 | N/A  |
| 8  | 7839  | KING   | 1981/11/17 | 1982-02-22 | N/A  |
| 9  | 7844  | TURNER | 1981/09/08 | 1981-12-14 | 0    |
| 10 | 7900  | JAMES  | 1981/12/03 | 1982-03-08 | N/A  |
| 11 | 7902  | FORD   | 1981/12/03 | 1982-03-08 | N/A  |
| 12 | 7934  | MILLER | 1982/01/23 | 1982-04-26 | N/A  |

6. 다음과 같은 결과가 나오도록 SQL문을 작성해 보세요.

사원들의 입사연도(HIRE\_YEAR)를 기준으로 부서별로 몇 명이 입사했는지 출력해 보세요.

```
SELECT TO_CHAR(HIREDATE, 'YYYY') AS HIRE_YEAR, DEPTNO, COUNT(*)AS CNT  
FROM EMP
```

```
GROUP BY TO_CHAR(HIREDATE, 'YYYY'), DEPTNO;
```



스크립트 출력 x | 질의 결과 x

SQL | 인출된 모든 행: 5(0.002초)

|   | HIRE_YEAR | DEPTNO | CNT |
|---|-----------|--------|-----|
| 1 | 1982      | 10     | 1   |
| 2 | 1980      | 20     | 1   |
| 3 | 1981      | 10     | 2   |
| 4 | 1981      | 20     | 2   |
| 5 | 1981      | 30     | 6   |

7. 다음과 같은 결과가 나오도록 SQL문을 작성해 보세요.

각 부서의 입사 연도별 사원 수, 최고 급여, 급여 합, 평균 급여를 출력하고 각 부서별 소개와 총계를 출력하세요.

```
SELECT DEPTNO, TO_CHAR(HIREDATE, 'YYYY') AS HIRE_YEAR,  
COUNT(*)AS CNT, MAX(SAL) AS MAX_SAL, SUM(SAL) AS SUM_SAL,  
AVG(SAL) AS AVG_SAL FROM EMP  
GROUP BY ROLLUP(DEPTNO, TO_CHAR(HIREDATE, 'YYYY'));
```

[illegible]

8. 모든 부서 정보와 사원 정보를 오른쪽과 같이 부서 번호, 사원 이름 순으로 정렬하여 출력해 보세요. (단, SQL-99 이전 방식과 SQL-99 방식을 각각 사용하여 작성하세요.)

## <SQL-99 이전 방식>

```
SELECT D.DEPTNO, D.DNAME, E.EMPNO, E.ENAME, E.JOB, E.SAL
FROM EMP E, DEPT D
WHERE E.DEPTNO(+) = D.DEPTNO
ORDER BY D.DEPTNO, E.ENAME;
```

## <SQL-99 방식>

```
SELECT D.DEPTNO, D.DNAME, E.EMPNO, E.ENAME, E.JOB, E.SAL
FROM EMP E RIGHT OUTER JOIN DEPT D ON (E.DEPTNO = D.DEPTNO)
ORDER BY D.DEPTNO, E.ENAME;
```

결과는 모두 같습니다.

| DEP... | DNAME         | EMPNO  | ENAME  | JOB       | SAL    |
|--------|---------------|--------|--------|-----------|--------|
| 1      | 10 ACCOUNTING | 7782   | CLARK  | MANAGER   | 2450   |
| 2      | 10 ACCOUNTING | 7839   | KING   | PRESIDENT | 5000   |
| 3      | 10 ACCOUNTING | 7934   | MILLER | CLERK     | 1300   |
| 4      | 20 RESEARCH   | 7902   | FORD   | ANALYST   | 3000   |
| 5      | 20 RESEARCH   | 7566   | JONES  | MANAGER   | 2975   |
| 6      | 20 RESEARCH   | 7369   | SMITH  | CLERK     | 800    |
| 7      | 30 SALES      | 7499   | ALLEN  | SALESMAN  | 1600   |
| 8      | 30 SALES      | 7698   | BLAKE  | MANAGER   | 2850   |
| 9      | 30 SALES      | 7900   | JAMES  | CLERK     | 950    |
| 10     | 30 SALES      | 7654   | MARTIN | SALESMAN  | 1250   |
| 11     | 30 SALES      | 7844   | TURNER | SALESMAN  | 1500   |
| 12     | 30 SALES      | 7521   | WARD   | SALESMAN  | 1250   |
| 13     | 40 OPERATIONS | (null) | (null) | (null)    | (null) |

9. 다음과 같이 모든 부서정보, 사원 정보, 급여 등급 정보, 각 사원의 직속 상관의 정보를 부서 번호, 사원 번호 순서로 정렬하여 출력해 보세요. (단, SQL-99 이전 방식과 SQL-99 방식을

각각 사용하여 작성하세요.)

#### <SQL-99 이전 방식>

```
SELECT D.DEPTNO, D.DNAME, E.EMPNO, E.ENAME, E.MGR, E.SAL, E.DEPTNO,  
S.LOSAL, S.HISAL, S.GRADE, E2.EMPNO AS MGR_EMPNO, E2.ENAME AS MGR_ENAME  
FROM DEPT D, EMP E, SALGRADE S, EMP E2  
WHERE D.DEPTNO = E.DEPTNO(+)  
AND E.SAL BETWEEN S.LOSAL(+) AND S.HISAL(+) AND E.MGR = E2.EMPNO(+)  
ORDER BY D.DEPTNO, E.EMPNO;
```

#### <SQL-99 방식>

```
SELECT D.DEPTNO, D.DNAME,  
       E.EMPNO, E.ENAME, E.MGR, E.SAL, E.DEPTNO,  
       S.LOSAL, S.HISAL, S.GRADE,  
       E2.EMPNO AS MGR_EMPNO, E2.ENAME AS MGR_ENAME  
FROM EMP E RIGHT OUTER JOIN DEPT D  
      ON (E.DEPTNO = D.DEPTNO)  
LEFT OUTER JOIN SALGRADE S  
      ON (E.SAL BETWEEN S.LOSAL AND S.HISAL)  
LEFT OUTER JOIN EMP E2  
      ON (E.MGR = E2.EMPNO)  
ORDER BY D.DEPTNO, E.EMPNO;
```

결과는 모두 같습니다.

| 스크립트 출력 x   질의 결과 x        |        |            |        |        |        |        |          |        |        |        |           |           |
|----------------------------|--------|------------|--------|--------|--------|--------|----------|--------|--------|--------|-----------|-----------|
| SQL   인출된 모든 행: 13(0.002초) |        |            |        |        |        |        |          |        |        |        |           |           |
|                            | DEPTNO | DNAME      | EMPNO  | ENAME  | MGR    | SAL    | DEPTNO_1 | LOSAL  | HISAL  | GRADE  | MGR_EMPNO | MGR_ENAME |
| 1                          | 10     | ACCOUNTING | 7782   | CLARK  | 7839   | 2450   | 10       | 2001   | 3000   | 4      | 7839      | KING      |
| 2                          | 10     | ACCOUNTING | 7839   | KING   | (null) | 5000   | 10       | 3001   | 9999   | 5      | (null)    | (null)    |
| 3                          | 10     | ACCOUNTING | 7934   | MILLER | 7782   | 1300   | 10       | 1201   | 1400   | 2      | 7782      | CLARK     |
| 4                          | 20     | RESEARCH   | 7369   | SMITH  | 7902   | 800    | 20       | 700    | 1200   | 1      | 7902      | FORD      |
| 5                          | 20     | RESEARCH   | 7566   | JONES  | 7839   | 2975   | 20       | 2001   | 3000   | 4      | 7839      | KING      |
| 6                          | 20     | RESEARCH   | 7902   | FORD   | 7566   | 3000   | 20       | 2001   | 3000   | 4      | 7566      | JONES     |
| 7                          | 30     | SALES      | 7499   | ALLEN  | 7698   | 1600   | 30       | 1401   | 2000   | 3      | 7698      | BLAKE     |
| 8                          | 30     | SALES      | 7521   | WARD   | 7698   | 1250   | 30       | 1201   | 1400   | 2      | 7698      | BLAKE     |
| 9                          | 30     | SALES      | 7654   | MARTIN | 7698   | 1250   | 30       | 1201   | 1400   | 2      | 7698      | BLAKE     |
| 10                         | 30     | SALES      | 7698   | BLAKE  | 7839   | 2850   | 30       | 2001   | 3000   | 4      | 7839      | KING      |
| 11                         | 30     | SALES      | 7844   | TURNER | 7698   | 1500   | 30       | 1401   | 2000   | 3      | 7698      | BLAKE     |
| 12                         | 30     | SALES      | 7900   | JAMES  | 7698   | 950    | 30       | 700    | 1200   | 1      | 7698      | BLAKE     |
| 13                         | 40     | OPERATIONS | (null) | (null) | (null) | (null) | (null)   | (null) | (null) | (null) | (null)    | (null)    |

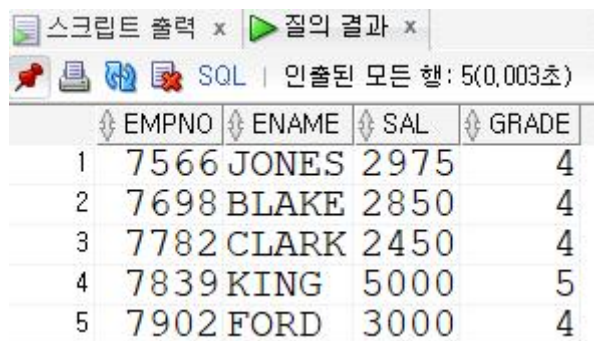
10. 직책이 SALESMAN인 사람들의 최고 급여보다 높은 급여를 받는 직원들의 직원 정보, 급여 등급 정보를 다음과 같이 출력하는 SQL문을 작성하세요(단, 서브쿼리를 활용할 때 다중행 함수를 사용하는 방법과 사용하지 않는 방법을 통해 직원 번호를 기준으로 오름차순으로 정렬하세요.)

```
SELECT E.EMPNO, E.ENAME, E.SAL, S.GRADE  
FROM EMP E, SALGRADE S
```

```

WHERE E.SAL BETWEEN S.LOSAL AND S.HISAL
AND SAL > (SELECT MAX(SAL)
           FROM EMP
           WHERE JOB = 'SALESMAN')
ORDER BY E.EMPNO;

```



|   | EMPNO | ENAME | SAL  | GRADE |
|---|-------|-------|------|-------|
| 1 | 7566  | JONES | 2975 | 4     |
| 2 | 7698  | BLAKE | 2850 | 4     |
| 3 | 7782  | CLARK | 2450 | 4     |
| 4 | 7839  | KING  | 5000 | 5     |
| 5 | 7902  | FORD  | 3000 | 4     |

수고하셨습니다.