**2019 클라우드 빅데이터 분석 및 시각화 실무 프로젝트 과정**

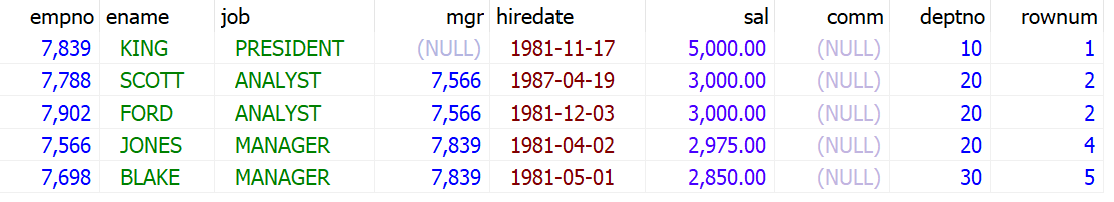
**DB SQL 시험**

**이름 : 고은경**

1. 시험문제 : 15문제
2. 시험시간 : 50분
3. 문제형태 : SQL문 실습 쿼리문 작성
4. 시험지 제출 : Word로 주어진 문제에 있는 내용을 쿼리문 작성을 통해 풀고, 문제를 푼 쿼리문을

Word 파일에 복사해서 넣고, 파일명을 **’2019\_빅데이터\_DBSQL\_홍길동.doc’** 파일로 저장해서

slack에 유현석 강사 개인톡으로 보내면 시험이 완료된 것 입니다.

1. 사원 테이블에서 각 사원에 급여(SAL)가 높은 순서대로 상위 5명을 아래 예제처럼 출력하세요.
   * ex)
   * 

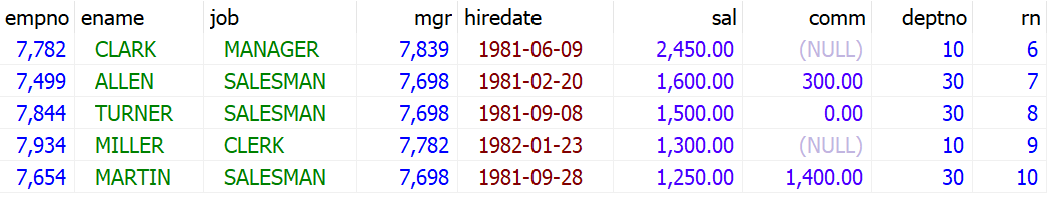
**SELECT \*, rank() over(ORDER BY sal desc) AS rownum**

**FROM emp**

**ORDER BY sal DESC**

**LIMIT 5**

**;**

1. 사원 테이블에서 각 사원에 급여(SAL)가 높은 순서대로 순위를 부여 했을 때 6등~10등인 사람을 순위대로 아래 예제처럼 출력하세요.
   * ex)
   * 

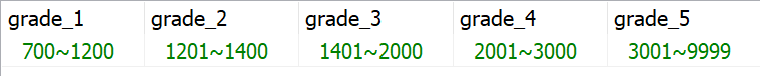
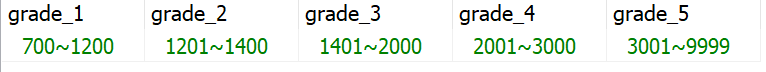
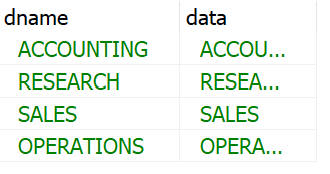
**SELECT \*, row\_number() over(ORDER BY sal desc) AS rn**

**FROM emp**

**ORDER BY sal DESC**

**LIMIT 5, 5**

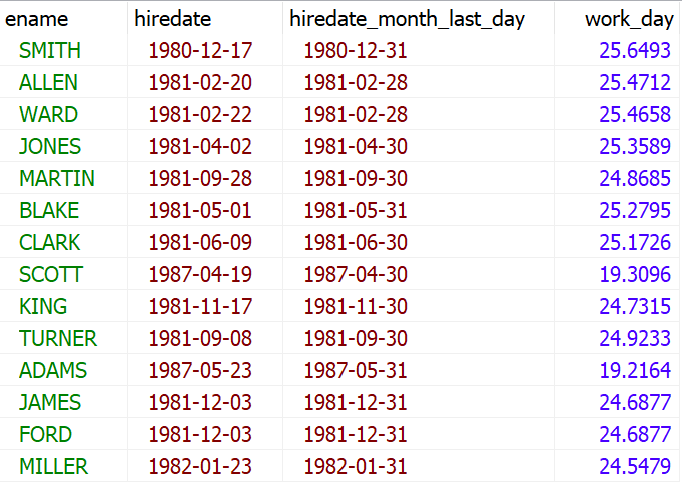
**;**

1. SALGRADE 테이블 데이터 세로 정보를 가로로 아래 예제처럼 출력하세요.
   * ex)
   * 
2. SALGRADE\_TEMP 테이블을 아래처럼 만들고, SALGRADE\_TEMP 테이블을 조회하면 아래 예제처럼 출력되게 SQL을 작성하세요.
   * ex) SELECT \* FROM SALGRADE\_TEMP  
     
3. 부서 테이블에서 DNAME 컬럼의 길이가 6자리 이상이면은 5자리까지 데이터만 보여주고 ‘..’ 뒤에 붙여준다. 아래 예제처럼 출력하세요
   * ex)
   * 

**SELECT dname, concat(SUBSTR(dname,1,5),'...') AS data**

**FROM dept**

**;**

1. 사원 테이블에서 고용일자(HIREDATE)의 해당 년월의 마지막 날짜(HIREDATE\_MONTH\_LASTDAY)를 구하고, 고용일자로 부터 2006/08/05일까지 근무한 일수(WORK\_DAY)를 아래처럼 출력하세요.
   * 

**SELECT ename,**

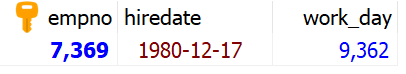
**hiredate,**

**LAST\_DAY(hiredate) AS hiredate\_month\_lastday,**

**DATEDIFF('2006-08-05',hiredate) AS workday**

**FROM emp**

**;**

1. 6번 예제에서 구한 각 사원에 근무일(WORK\_DAY)이 가장 높은 사람을 한 명을 아래 예제처럼 출력하세요
   * ex)
   * 

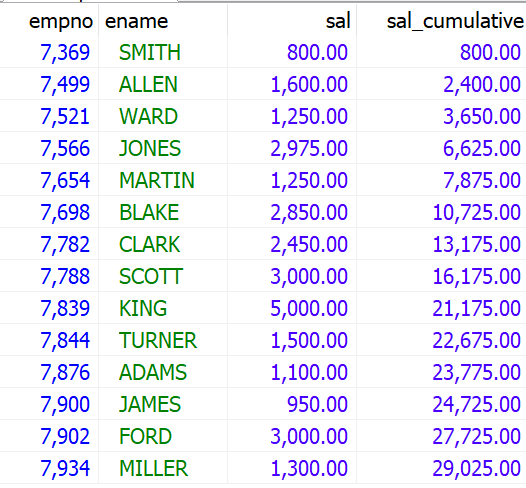
**SELECT empno,**

**hiredate,**

**MAX(DATEDIFF('2006-08-05',hiredate)) AS work\_day**

**FROM emp**

**;**

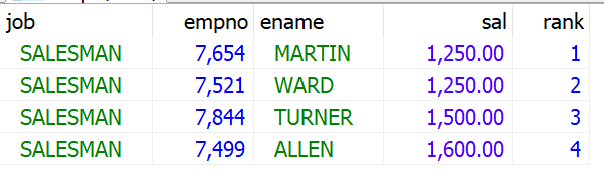
1. 사원 테이블에서 급여(SAL) 값을 누계(SAL\_CUMULATIVE) 값으로 사원번호(EMPNO) 오름차순 정렬 순서로 아래 예제처럼 출력하세요.
   * ex)
   * 

**SELECT empno, ename, sal, sum(sal) over(order by empno) 누계**

**FROM emp**

**ORDER BY empno**

**;**

1. 사원 테이블에서 직업이 ‘SALESMAN’ 사원 중에 급여(SAL) 낮은 순서대로 순위(RANK)를 아래 예제처럼 출력하세요
   * ex) 급여가 같은 경우 데이터 정렬 순서 나오는 대로 순위 정함.
   * 

**SELECT job,**

**empno,**

**ename,**

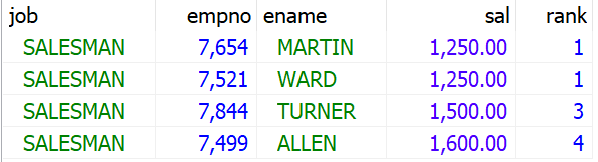
**sal,**

**row\_number() over(order by sal) AS rank**

**FROM emp**

**WHERE job = 'SALESMAN'**

**;**

1. 사원 테이블에서 직업이 ‘SALESMAN’ 사원 중에 급여(SAL) 낮은 순서대로 순위(RANK)를 아래 예제처럼 출력하세요?
   * ex) 급여가 같은 경우 순위는 같고 동률 순위 사람 수 만큼 다음 순위에 순위 차를 둠
   * 

**SELECT job,**

**empno,**

**ename,**

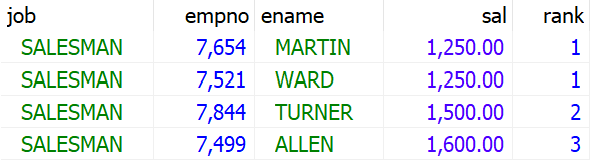
**sal,**

**rank() over(order by sal) AS rank**

**FROM emp**

**WHERE job = 'SALESMAN'**

**;**

1. 사원 테이블에서 직업이 ‘SALESMAN’ 사원 중에 급여(SAL) 낮은 순서대로 순위(RANK)를 아래 예제처럼 출력하세요
   * ex) 급여가 같은 경우 순위는 같고 다음 순위는 위 순위 번호 다음값
   * 

**SELECT job,**

**empno,**

**ename,**

**sal,**

**dense\_rank() over(order by sal) AS rank**

**FROM emp**

**WHERE job = 'SALESMAN'**

**;**

1. EMP테이블을 백업 하기 위하여(복제하기 위하여) 한 sql문장으로 데이터를 EMP\_BAK TABLE을 생성하여 넣는 문장을 작성하시오
   * (테이블 생성 및 INSERT가 한 문장으로 수행 되어야 합니다.)

**CREATE emp\_bak(**

**INSERT \* FROM emp**

**)ENGINE=INNODB;**

**COMMIT;**

1. 사원(EMP이름)테이블에서 직업(JOB)이 ‘SALESMAN’ 인 사원 급여(SAL)에 400 더하는 수정(UPDATE) 구문을 구하세요

**UPDATE emp SET sal = sal + 400 WHERE job = 'SALESMAN';**

**COMMIT;**

1. 사원(EMP이름)테이블에서 급여(SAL)가 사원 평균급여 보다 높은 사원을 대상으로 고용일자(HIREDATE)를 1년 더하는 수정(UPDATE) 구문을 구하세요
2. 사원(EMP이름)테이블에서 전체 사원을 대상으로 COMM 컬럼에 100 을 더하고 직업(JOB)이 ‘CLERK’ 인 사원은 현 급여에서 2배, ‘MANAGER’ 인 직업을 가진 사원은 현 급여에서 3배, 이외 직업을 가진 사원은 현 급여에서 4배를 더하는 수정(UPDATE) 구문을 구하세요