|  |
| --- |
| 1.    각 부서명과 부서별 직원 수를 파악하기위해서는 employee 테이블과 department 테이블을 Join해주어야 한다. employee의 dno와 department의 dno가 두 테이블에 공통적으로 존재함으로 컬럼의 값이 일치되는 행을 연결하여 결과를 생성하는 조인의 방법인 Equi Join을 사용한다. (where절에 e.dno = d.dno으로 두 테이블 조인) 부서별 직원의 수를 출력하기 위해서 조인된 테이블을 부서명(dname)으로 묶어줌으로써 부서별 직원으로 그룹이 묶이게 된다. 이제 부서별 직원의 수를 출력을 해주기 위해서 count(dname)를 이용하여 결과값을 뽑아낸다. 여기서 부서명(dname)도 출력을 할 수 있는데 단순컬럼임에도 출력할 수 있는 이유는 group by로 묶인 단순컬럼이기 때문이다. |
| 2.    각 부서명을 출력하기 위해서 employee테이블과 department테이블의 공통 컬럼값인 dno으로 Equi Join (where절에 e.dno = d.dno으로 두 테이블 조인)해주고 부서별 직원들의 평균 급여등급을 출력하기 위해서 employee테이블과 salgrade테이블을 조인해주어야한다. 급여에 따른 등급을 매겨야하기 때문에 Non-Equi Join을 이용하여 where절에 salary >= lowsal and salary <= highsal 을 조건으로 이용하여 salary가 lowsal과 highsal 사이의 값에 있을 때 등급을 매겨 테이블을 조인한다. 총 3개의 테이블이 Join된 테이블에서 부서명(dname)으로 묶어준다. 부서별로 묶인 직원들의 급여등급의 평균값을 내주기 위해서 avg(s.grade)를 사용한다. 부서명을 나타내기 위해서 부서명도 같이 출력해주는데 부서명은 단일 컬럼인데도 출력 가능한 이유는 이유는 group by로 묶인 단순컬럼이기 때문이다. |
| 3.    사원의 소속 부서명을 알기 위해서 employee테이블과 department테이블을 Equi Join(e.dno = d.dno)을 이용해 테이블을 조인하여 사원별 소속부서를 Join해준다. 급여등급을 알기위해서는 employee테이블과 salgrade테이블을 Non-Equi Join(e.salary between s.lowsal and s.highsal)을 이용해 사원별 급여등급을 연결해준다. 급여에 따른 순위를 출력하는 것은 dense\_rank를 이용해서 급여 등급의 데이터 값에 따른 오름차순으로 순위를 매겨준다. |
| 4.    중복제거를 해주는 집계함수 distinct를 이용하여 employee테이블의 salary 출력시 중복을 제거하고 출력해준다. |
| 5.    최소값을 구하는 집계함수 min()을 이용하여 employee테이블의 hiredate 칼럼 값에서 가장 오래 입사일 데이터를 출력해준다.(날짜는 min()일 때는 가장 늦은 날짜가 나오고, max()일 때는 가장 빠른 날짜가 나온다.) |
| 6.    최소값을 구하는 집계함수 min()을 이용하여 employee 테이블의 salary컬럼의 가장 적은 급여 데이터 값을 출력해준다. |
| 7.    문제에서 요구하는 컬럼값을 구하기 위해서는 세개의 테이블을 조인하여야 한다. employee테이블과 department테이블을 Equi Join(e.dno = d.dno)을 이용해 테이블을 조인하고 급여등급을 알기위해서는 employee테이블과 salgrade테이블은 Non-Equi Join(e.salary between s.lowsal and s.highsal)을 이용해 조인을 한다. ‘소속 직원 수’를 구하기위해 dno를 그룹화 시킨후에 count()함수를 이용해서 구한다. ‘소속 직원 평균 급여’와 ‘급여 합’, ‘급여 등급’을 구하기 위해서는 집계함수 avg()와 sum(), avg()를 이용하여 출력해준다. 부서명과 부서위치는 단순컬럼이기 때문에 출력해주기 위해서 group by를 이용하여 묶어주면 단순컬럼임에도 불구하고 출력할 수 있기에그룹화 시켜준다. |
| 8.    테이블 세 개를 Join해야 문제에서 원하는 컬럼 값을 출력할 수 있다. 여기서 세 개의 데이터를 Join하기 위해서 각 테이블의 공통적으로 존재하는 컬럼의 값을 찾아야한다. instructor테이블과 teaches 테이블의 공통 컬럼은 ID이다. 따라서 i.id = t.id를 하여 instructor 테이블과 teaches 테이블을 조인해준다. teaches테이블과 course테이블의 공통 걸럼은 course\_id 이다. 따라서 t.course\_id = c.course\_id를 이용해서 두 테이블을 조인해준다. 이로써 세 개의 테이블이 조인되었고 문제에서 원하는 교수님의 성함과 학과, 강의 이름과 이수학점, 개설 년도와 계절 정보를 select를 이용해서 출력해주면 된다. |