

Quiz

4-1-1. locations 테이블에는 전 세계에 있는 지역 사무소 주소 정보가 나와 있습니다. 각 국가별로 지역사무소가 몇 개나 되는지 찾는 쿼리를 작성해 보세요.

정답 :

```
SELECT country_id, COUNT(*)  
  FROM locations  
 GROUP BY country_id  
 ORDER BY country_id;
```

Quiz

4-1-2. employees 테이블에서 년도에 상관 없이 분기별로 몇 명의 사원이 입사했는지 구하는 쿼리를 작성해 보세요.

정답 :

```
SELECT TO_CHAR(hire_date, 'Q'), COUNT(*)  
FROM employees  
GROUP BY TO_CHAR(hire_date, 'Q')  
ORDER BY 1;
```

Quiz

4-1-3. 다음 쿼리는 employees 테이블에서 job_id별로 평균 급여를 구한 것인데, 여기서 평균을 직접 계산하는 avg_salary1 이란 가상컬럼을 추가해 보세요.

(평균 = 총 금액 / 인원수)

```
SELECT job_id, ROUND(AVG(salary),0) avg_salary  
FROM employees  
GROUP BY job_id  
ORDER BY 1;
```

정답 :

```
SELECT job_id, ROUND(AVG(salary),0) avg_salary,  
ROUND(SUM(salary) / COUNT(*), 0)  
FROM employees  
GROUP BY job_id  
ORDER BY 1;
```

Quiz

4-2-1. 다음 쿼리를 실행하면 오류가 발생하는데 그 이유는 무엇일까요?

```
SELECT job_id jobs  
  FROM employees  
 WHERE department_id = 60  
UNION  
SELECT job_id  
  FROM employees  
 WHERE department_id = 90  
ORDER BY job_id;
```

정답 :

첫 번째 쿼리의 select 절에서 job_id의 alias를 jobs로 주었기 때문에 맨 끝에 있는 order by 절에서 job_id 대신 jobs를 명시해야 합니다.

Quiz

4-2-2. 집합 연산자를 사용해 employees 테이블에서 2001과 2003년에 입사한 사원의 사원번호와 입사일자를 조회하는 쿼리를 작성해 보세요.

정답 :

```
SELECT employee_id, hire_date
  FROM employees
 WHERE TO_CHAR(hire_date, 'YYYY') = '2001'
 UNION ALL
SELECT employee_id, hire_date
  FROM employees
 WHERE TO_CHAR(hire_date, 'YYYY') = '2003'
 ORDER BY 2;
```

Quiz

4-2-3. employees 테이블에서 job_id 별로 급여(salary)의 합계를 구하고, 마지막에 전체 급여 합계를 구하는 쿼리를 UNION 연산자를 사용해 작성해 보세요.

정답 :

```
SELECT job_id, SUM(salary)
FROM employees
GROUP BY job_id
UNION
SELECT '합계', SUM(salary)
FROM employees
ORDER BY 1 ;
```