**SQL Test**

**(이름.sq으로 파일저장)**

**※ 컬럼명과 데이터 정렬 순서는 결과화면과 동일하게 나타나야 합니다.**

**(컬럼명 다른 경우 -1점, 정렬순서가 다른 경우 -2점)**

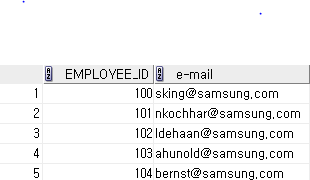
Q1. EMPLOYEES 테이블에 저장되어있는 직원들 중 SALARY가 10,000 이상인 사람들을 JOB\_ID, EMPLOYEE\_ID 오름차순으로 정렬하시오.





(인출된 모든 행: 19)

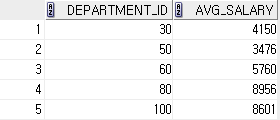
Q2. EMPLOYEES 테이블에 저장되어있는 직원들의 EMPLOYEE\_ID와 EMAIL을 다음의 출력결과와 같은 형태로 나타내고, EMPLOYEE\_ID 오름차순으로 정렬하시오. 이메일 주소는 소문자에 뒤에 '@samsung.com' 이 붙고, 컬럼명은 e-mail 로 한다.





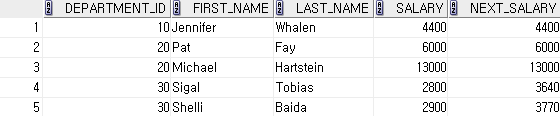
(인출된 모든 행: 107)

Q3. EMPLOYEES 테이블에 저장되어있는 직원들을 DEPARTMENT\_ID별로 구분하고, 구분된 각 부서 별로 평균 SALARY를 구하시오. 단, 직원들의 수가 5명 이상인 부서만 나타나도록 출력하고, 평균 SALARY는 반올림하여 정수로 표현하시오. 정렬은 DEPARTMENT\_ID 오름차순으로 정렬하시오.



(인출된 모든 행: 5)

Q4. EMPLOYEES 테이블에 저장되어있는 직원들의 NEXT\_SALARY를 구하고자 한다. DEPARTMENT\_ID가 30인 직원들은 SALARY의 1.3배, 50인 직원들은 1.5배, 80인 직원들은 0.9배, 그 외에는 기존 SALARY를 그대로 NEXT\_SALARY에 출력되도록 하시오. 정렬은 DEPARTMENT\_ID기준 오름차순, FIRST\_NAME기준 내림차순으로 정렬하시오.





(인출된 모든 행: 107)

Q5. EMPLOYEES 테이블에 저장되어있는 직원들 중 SALARY가 가장 높은 직원의 정보를 다음과 같은 형태로 조회하시오.



(인출된 모든 행: 1)

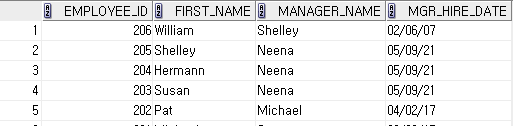
Q6. EMPLOYEES, JOBS 테이블을 활용하여 직원들의 JOB\_TITLE을 포함한 정보를 조회하고자 한다. 단, JOB\_TITLE에 'Manager'가 포함된 직원들만 EMPLOYEE\_ID 내림차순으로 조회하시오.





(인출된 모든 행: 14)

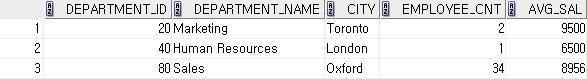
Q7. EMPLOYEES 테이블에 저장되어있는 직원들의 정보와 직원들을 관리하는 MANAGER에 대한 정보를 다음과 같이 조회하시오. 이 때 MANAGER\_ID가 없는 경우에도 EMPLOYEE\_ID와 직원의 FIRST\_NAME은 출력되도록 하시오. (MANAGER\_NAME은 MANAGER의 FIRST\_NAME이고, MGR\_HIRE\_DATE는 MANAGER의 HIRE\_DATE이다. EMPLOYEE\_ID기준 내림차순으로 정렬하시오.)





(인출된 모든 행: 107)

Q8. EMPLOYEES, DEPARTMENTS, LOCATIONS 테이블을 활용하여, 다음과 같은 정보들을 조회하려고 한다. 먼저 EMPLOYEES 테이블에 저장되어 있는 직원들을 DEPARTMENT\_ID별로 나누고, 그에 해당하는 DEPARTMENT\_NAME과 CITY정보를 출력한다. 그리고 나눠진 DEPARTMENT\_ID별로 속해있는 직원의 수를 카운트하고, 직원들의 평균SALARY를 반올림하여 나타내시오. 단, CITY가 'Toronto', 'London', 'Oxford'인 부서만 조회하고, DEPARTMENT\_ID 오름차순으로 정렬하시오.



(인출된 모든 행: 3)

Q9. JOBS 테이블에 저장되어있는 JOB관련된 정보 중 JOB\_ID가 'IT\_PROG'인 데이터를 다음과 같이 수정하시오. JOB\_TITLE을 Solution Programmer로, MAX\_SALARY를 50000으로 UPDATE하시오.



**↓**



**문제10. 매개변수로 받은 직책(job\_title)과 급여범위(loSal, hiSal)에 해당하는 직원의 이름(first\_name), 급여(salary), 직급(job\_title) 목록을 출력하는 printEmployeeList() 메소드를 구현하시오.**

|  |
| --- |
| import java.sql.Connection;  import java.sql.DriverManager;  public class exam10{    public static void main(String[] args) throws Exception {  printEmployeeList("mana", 10000, 15000);  }  private static void printEmployeeList(String jobTtile, int loSal, int hiSal)  throws Exception {  Connection conn =DriverManager.getConnection("jdbc:oracle:thin:@127.0.0.1:1521:xe","hr", "hr");    // 구현하시오.    }  } |

**<< 제약조건 >>**

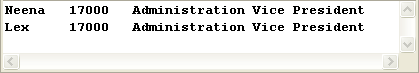
인자로 주어지는 직급(job\_title)은 대소문자 구분 없이 입력되며, 직급의 일부만 입력되는 것을 가정한다. 예를 들어 “Manager” 는 “mana” 형태 만으로 입력될 수 있다.

**<< 실행결과 >>**

printEmployeeList("mana", 10000, 15000); 형태로 호출했을 경우



printEmployeeList("admin", 15000, 20000); 형태로 호출했을 경우



**<< 채점기준 >>**

1. JDBC Driver를 찾지 못해 ClassNotFoundException이 발생하는 경우는 오답으로 처리한다.

2. 컴파일이 안되거나 예외가 발생되는 경우 오답으로 처리한다.