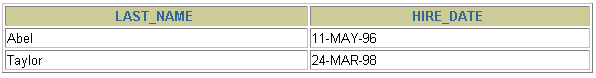
**서브쿼리 문제**

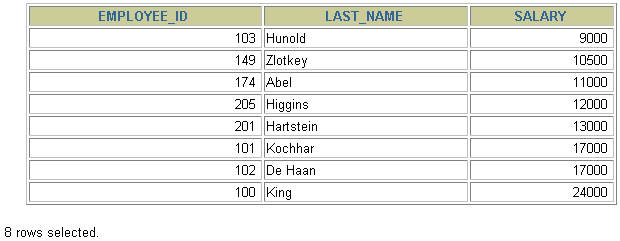
1. The HR department needs a query that prompts the user for an employee last name. The query then displays the last name and hire date of any employee in the same department as the employee whose name they supply (excluding that employee). For example, if the user enters Zlotkey, find all employees who work with Zlotkey (excluding Zlotkey).

Zlotkey사원과 같은 부서에 근무하는 사원 데이터 검색 (Zlotkey사원은 제외)



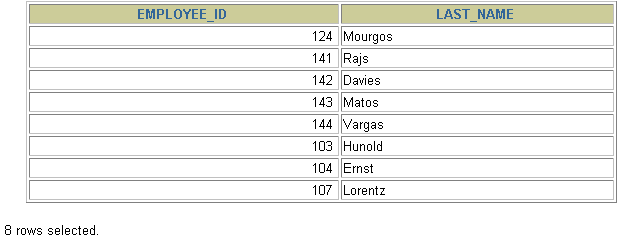
2. Create a report that displays the employee number, last name, and salary of all employees who earn more than the average salary. Sort the results in order of ascending salary.

전체 사원의 평균월급보다 월급을 많이 받는 사원 검색 (월급의 내림차순으로 정렬)



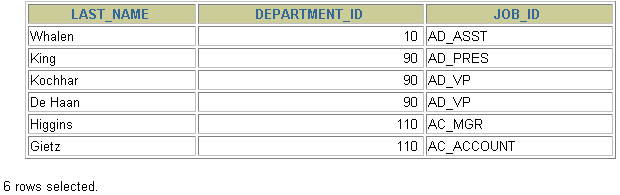
3. Write a query that displays the employee number and last name of all employees who work in a department with any employee whose last name contains a *u*. Place your SQL statement in a text file named lab\_06\_03.sql. Run your query.

last\_name에 'u' 문자가 포함되어 있는 사원들과 같은 부서에서 일하는 사원 검색



4. The HR department needs a report that displays the last name, department number, and job ID of all employees whose department location ID is 1700.

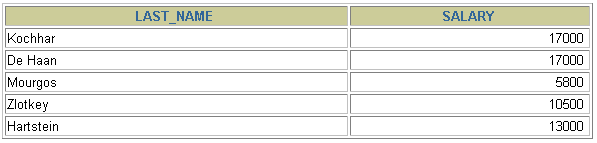
location ID 가 1700인 부서에 근무하는 사원 검색



Modify the query so that the user is prompted for a location ID. Save this to a file named lab\_06\_04.sql.

5. Create a report for HR that displays the last name and salary of every employee who reports to King.

King사원에게 직접 보고하는 모든 사원 검색(King사원을 관리자로 가지는 모든 사원 검색)



6. Create a report for HR that displays the department number, last name, and job ID for every employee in the Executive department.

'Executive' 부서에 근무하는 사원 검색



7. Modify the query in lab\_06\_03.sql to display the employee number, last name, and salary of all employees who earn more than the average salary and who work in a department with any employee whose last name contains a *u*. Resave lab\_06\_03.sql as lab\_06\_07.sql. Run the statement in lab\_06\_07.sql.

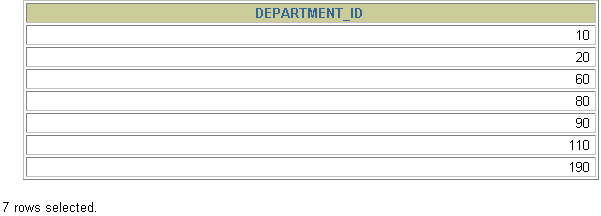
**lastname에 'u'가 포함된 사원들과 같은 부서에 근무하면서 전체 사원의 평균월급보다 월급을 많이 받는 사원 검색**



**집합연산자 실습]**

1. The HR department needs a list of department IDs for departments that do not contain the job ID ST\_CLERK. Use set operators to create this report.

job ID가 'ST\_CLERK' 인 직무가 존재하지 않는 부서를 검색



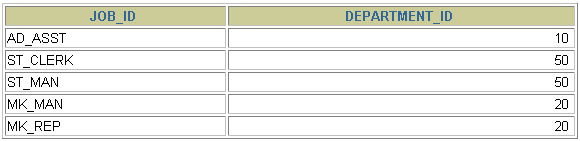
2. The HR department needs a list of countries that have no departments located in them. Display the country ID and the name of the countries. Use set operators to create this report.

부서가 없는 country ID와 Name 출력 (countries와 locations 테이블 참조)

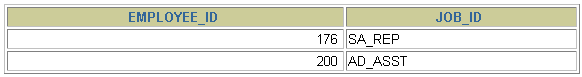


3. Produce a list of jobs for departments 10, 50, and 20, in that order. Display job ID and department ID using set operators.

set opreator를 사용해서 10, 50, 20 부서번호 순으로 JOB ID와 Department ID 순으로 출력



4. Create a report that lists the employee IDs and job IDs of those employees who currently have a job title that is the same as their job title when they were initially hired by the company (that is, they changed jobs but have now gone back to doing their original job).



The HR department needs a report with the following specifications:

* + - * Last name and department ID of all the employees from the EMPLOYEES table, regardless of whether or not they belong to a department
      * Department ID and department name of all the departments from the DEPARTMENTS table, regardless of whether or not they have employees working in them

Write a compound query to accomplish this.



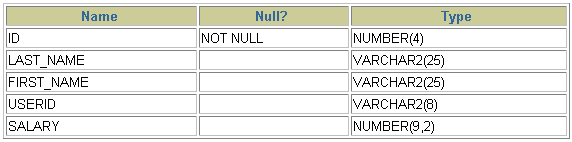
**DML실습 (insert, update, delete)**

The HR department wants you to create SQL statements to insert, update, and delete employee data. As a prototype, you use the MY\_EMPLOYEE table, before giving the statements to the HR department.

Insert data into the MY\_EMPLOYEE table.

1. Run the statement in the lab\_08\_01.sql script to build the MY\_EMPLOYEE table to be used for the lab.

2. Describe the structure of the MY\_EMPLOYEE table to identify the column names.

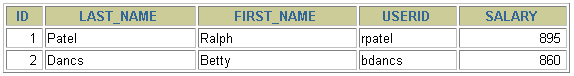


3. Create an INSERT statement to add *the first row* of data to the MY\_EMPLOYEE table from the following sample data. Do not list the columns in the INSERT clause. *Do not enter all rows yet.*



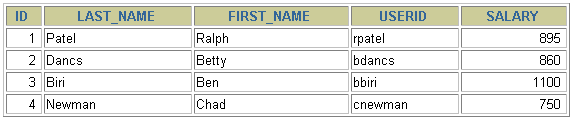
4. Populate the MY\_EMPLOYEE table with the second row of sample data from the preceding list. This time, list the columns explicitly in the INSERT clause.

5. Confirm your addition to the table.



6. Write an insert statement in a dynamic reusable script file named loademp.sql to load rows into the MY\_EMPLOYEE table. Concatenate the first letter of the first name and the first seven characters of the last name to produce the user ID. Save this script to a file named lab\_08\_06.sql.

7. Populate the table with the next two rows of sample data listed in step 3 by running the insert statement in the script that you created.

8. Confirm your additions to the table.  
 

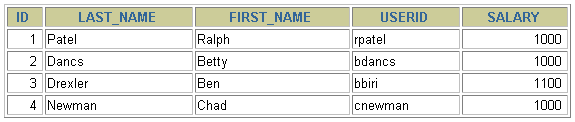
9. Make the data additions permanent.

Update and delete data in the MY\_EMPLOYEE table.

10. Change the last name of employee 3 to Drexler.

11. Change the salary to $1,000 for all employees who have a salary less than $900.

12. Verify your changes to the table.



13. Delete Betty Dancs from the MY\_EMPLOYEE table.

14. Confirm your changes to the table.

