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).

**UNDEFINE name**

**SELECT last\_name, hire\_date**

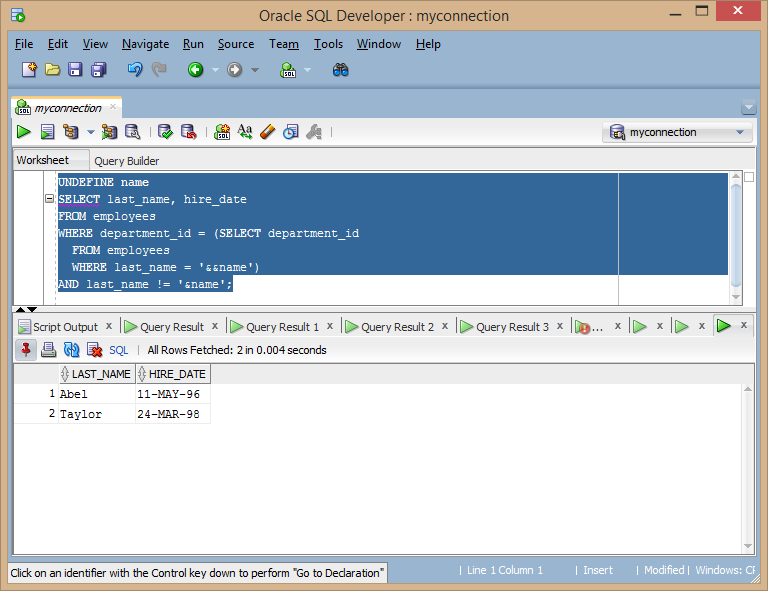
**FROM employees**

**WHERE department\_id = (SELECT department\_id**

**FROM employees**

**WHERE last\_name = '&&name')**

**AND last\_name != '&name';**



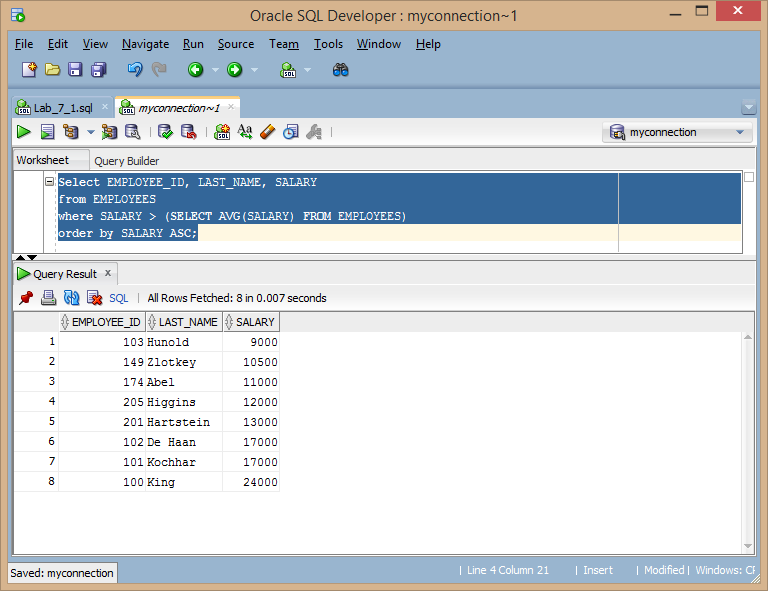
2) Create a report that displays the employee number, last name, and salary of all employees who earn more than the average salary. Sort the results in order of ascending salary.

**Select EMPLOYEE\_ID, LAST\_NAME, SALARY**

**from EMPLOYEES**

**where SALARY > (SELECT AVG(SALARY) FROM EMPLOYEES)**

**order by SALARY ASC;**



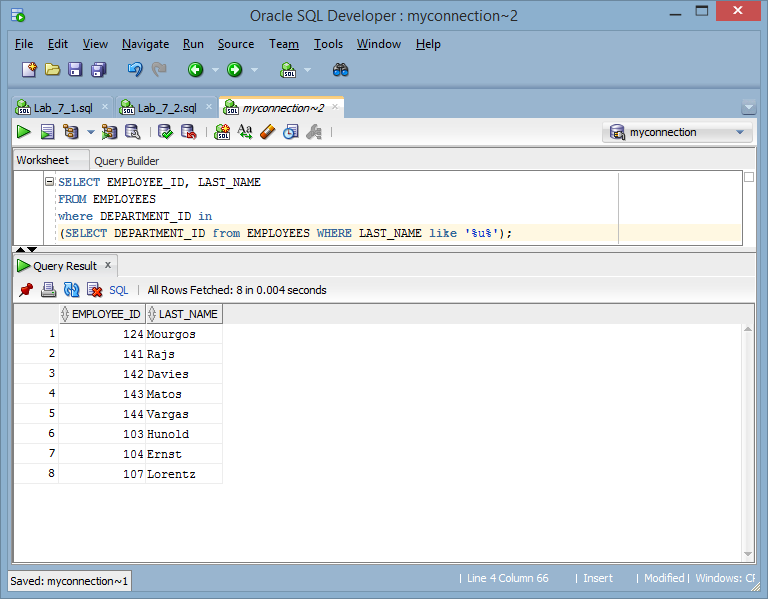
3) Write a query that displays the employee number and last name of all employees who work in a department with any employee whose last name contains the letter “u.” Save your SQL statement as lab\_07\_03.sql. Run your query.

**SELECT EMPLOYEE\_ID, LAST\_NAME**

**FROM EMPLOYEES**

**where DEPARTMENT\_ID in**

**(SELECT DEPARTMENT\_ID from EMPLOYEES WHERE LAST\_NAME like '%u%');**



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.

**SELECT e.LAST\_NAME, e.DEPARTMENT\_ID, e.JOB\_ID**

**FROM EMPLOYEES e join DEPARTMENTS d on e.DEPARTMENT\_ID = d.DEPARTMENT\_ID**

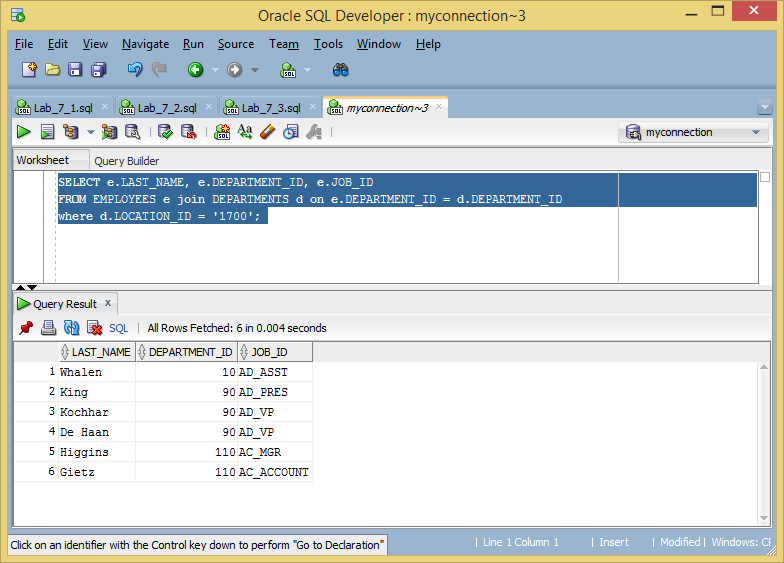
**where d.LOCATION\_ID = '1700';**

Modify the query so that the user is prompted for a location ID. Save this to a file named lab\_07\_04.sql.

**SELECT e.LAST\_NAME, e.DEPARTMENT\_ID, e.JOB\_ID**

**FROM EMPLOYEES e join DEPARTMENTS d on e.DEPARTMENT\_ID = d.DEPARTMENT\_ID**

**where d.LOCATION\_ID = '&Department';**



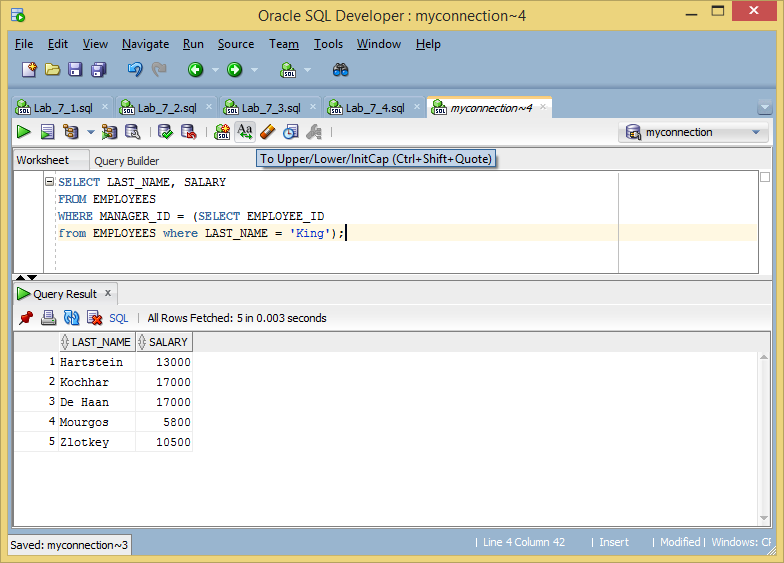
5) Create a report for HR that displays the last name and salary of every employee who reports to King.

**SELECT LAST\_NAME, SALARY**

**FROM EMPLOYEES**

**WHERE MANAGER\_ID = (SELECT EMPLOYEE\_ID**

**from EMPLOYEES where LAST\_NAME = 'King');**



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

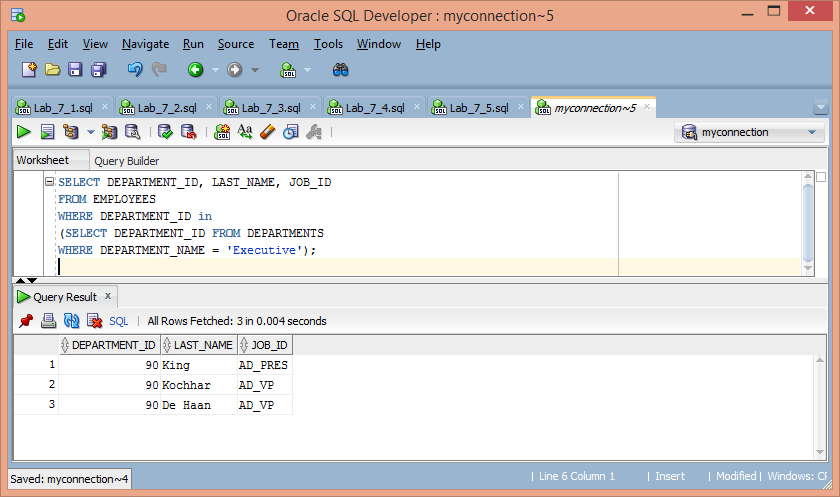
**SELECT DEPARTMENT\_ID, LAST\_NAME, JOB\_ID**

**FROM EMPLOYEES**

**WHERE DEPARTMENT\_ID in**

**(SELECT DEPARTMENT\_ID FROM DEPARTMENTS**

**WHERE DEPARTMENT\_NAME = 'Executive');**

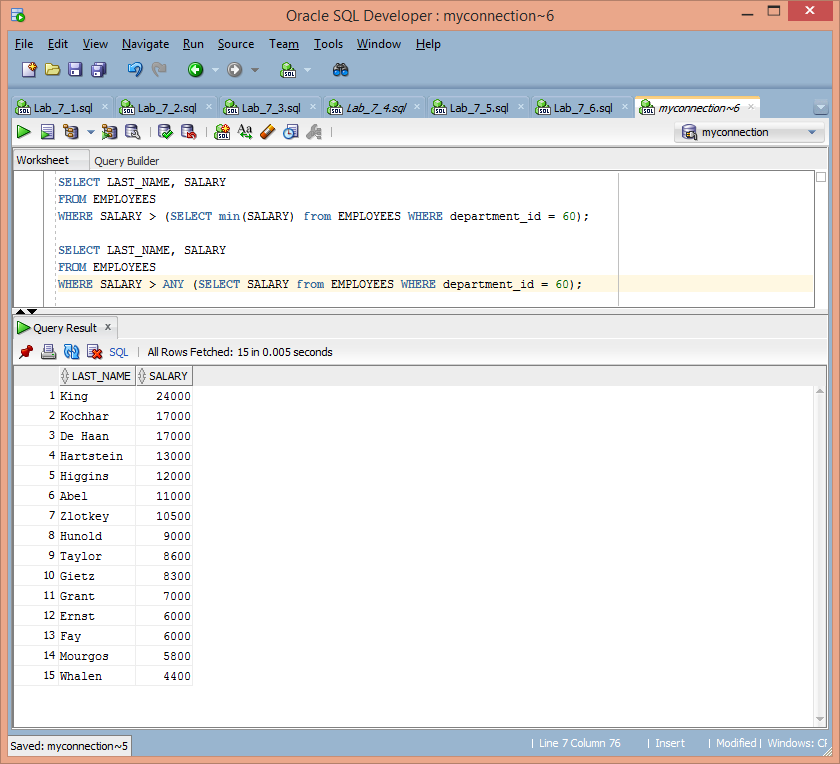


7) Create a report that displays a list of all employees whose salary is more than the salary of any employee from department 60.

**SELECT LAST\_NAME, SALARY**

**FROM EMPLOYEES**

**WHERE SALARY > ANY (SELECT SALARY from EMPLOYEES WHERE department\_id = 60);**



8) Modify the query in lab\_07\_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.” Save lab\_07\_03.sql as lab\_07\_08.sql again. Run the statement in lab\_07\_08.sql.

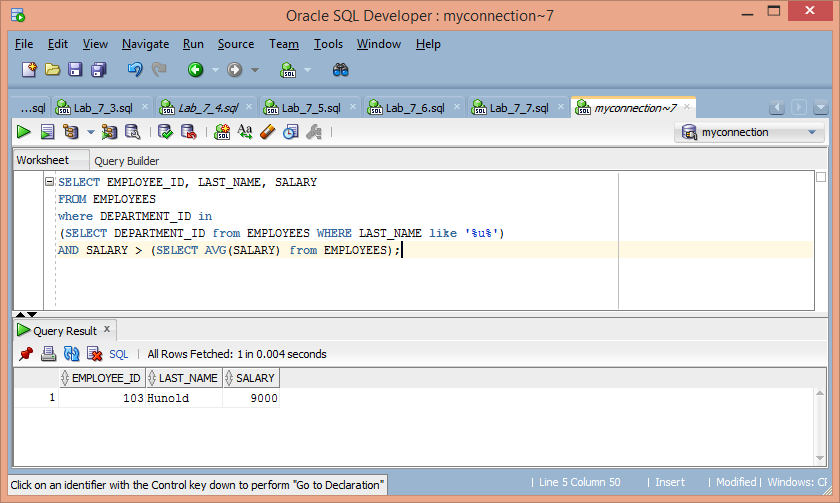
**SELECT EMPLOYEE\_ID, LAST\_NAME, SALARY**

**FROM EMPLOYEES**

**where DEPARTMENT\_ID in**

**(SELECT DEPARTMENT\_ID from EMPLOYEES WHERE LAST\_NAME like '%u%')**

**AND SALARY > (SELECT AVG(SALARY) from EMPLOYEES);**



9) What are the primary properties of subqueries? When are subqueries necessary as opposed to alternative query approaches?

**A subquery is a query nested inside another query. Subqueries can be necessary when the tables in question do not necessarily have a direct relationship.**

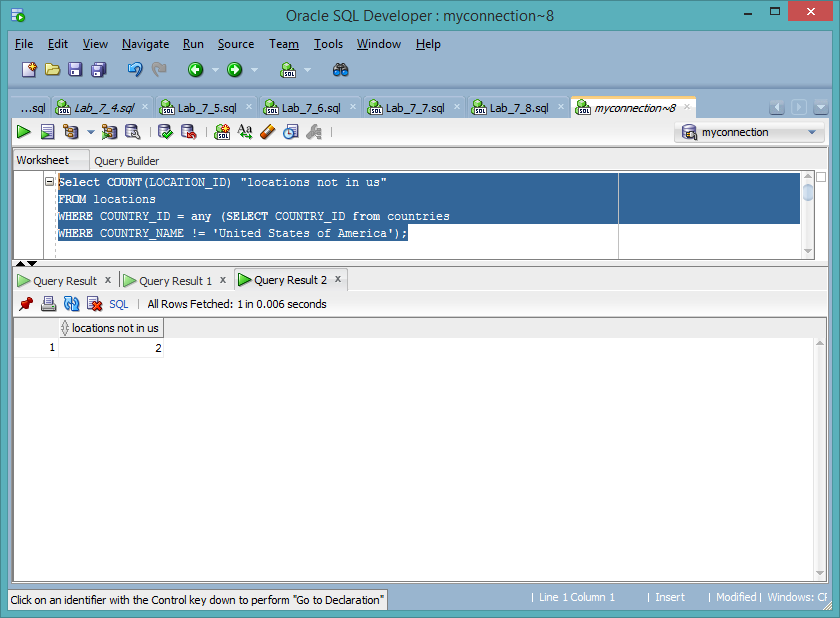
10) Using justification of a viable business report, create two (2) useful subqueries using the minimum entities of regions, countries, locations, and departments. Provide two (2) screenshots that depict each subquery.

**Select COUNT(LOCATION\_ID) "locations not in us"**

**FROM locations**

**WHERE COUNTRY\_ID = any (SELECT COUNTRY\_ID from countries**

**WHERE COUNTRY\_NAME != 'United States of America');**



**SELECT DEPARTMENT\_NAME "Departments not in the US"**

**FROM DEPARTMENTS**

**WHERE LOCATION\_ID = any**

**(SELECT LOCATION\_ID FROM LOCATIONS**

**WHERE COUNTRY\_ID = any**

**(SELECT COUNTRY\_ID FROM COUNTRIES**

**WHERE country\_name != 'United States of America'));**

